

Yandex



Android UI Components

Stepan Churyukanov

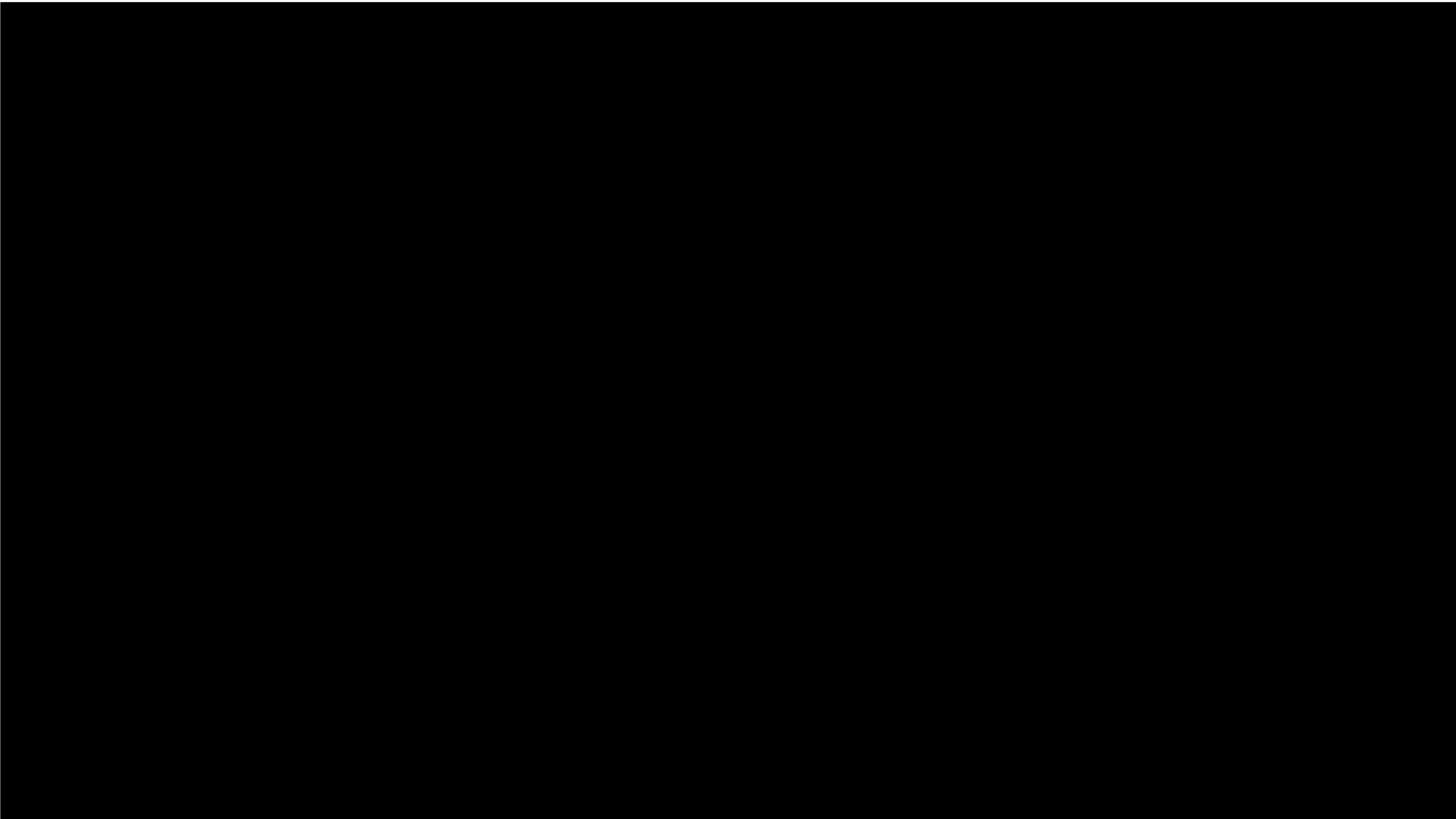
Table of contents

1	<u>Evolution Of Android</u>	4	<u>Navigation Component & Drawer</u>
2	<u>Material Design</u>	5	<u>App Bar scrollable regions</u>
3	<u>Snackbar</u>	6	<u>RecyclerView</u>

Evolution Of Android



Evolution Of Android



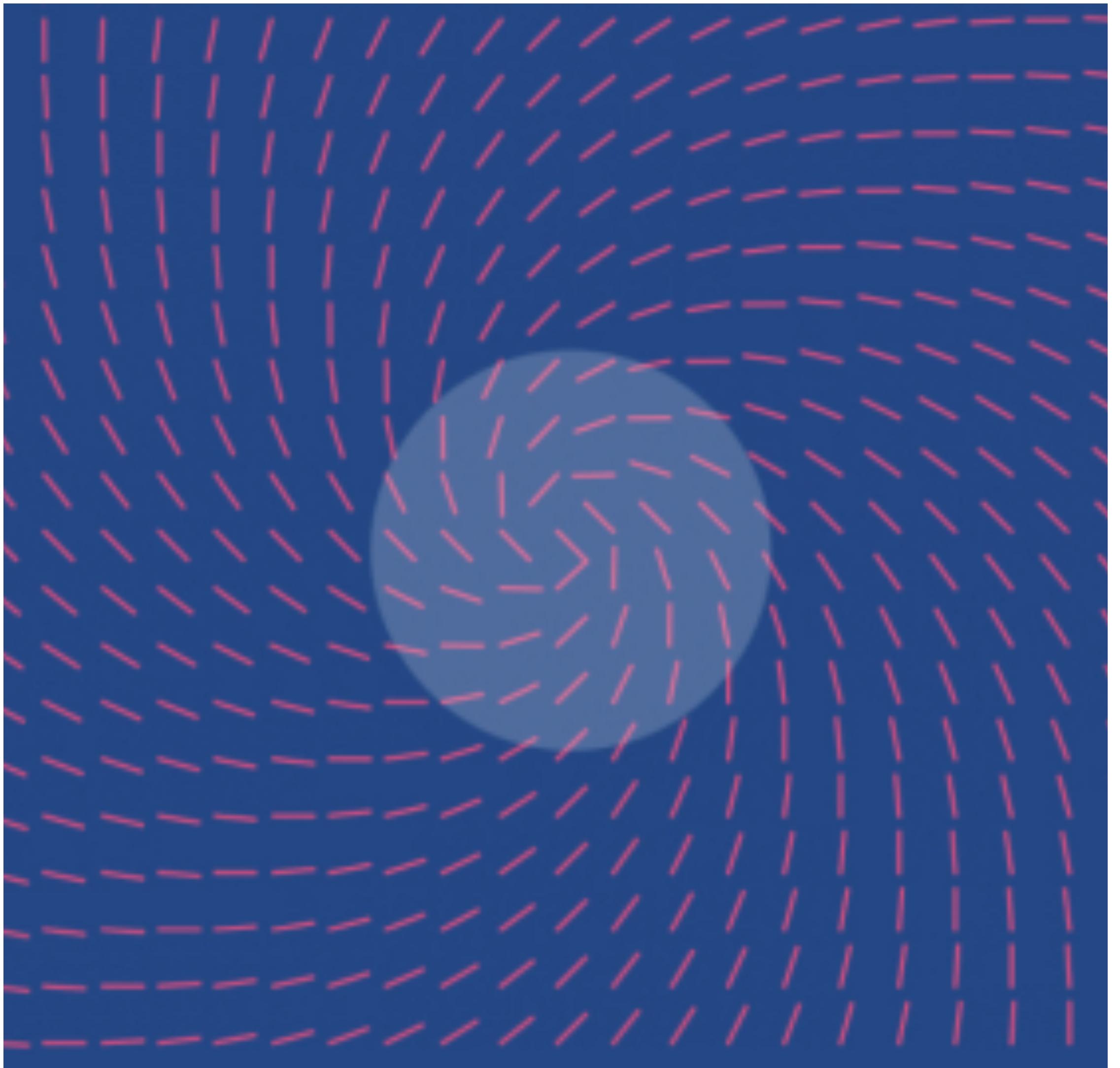
Material Design



Goals

Цели

- Разработать основополагающую систему, предоставляющую единое восприятие на платформах и устройствах всевозможных размеров.



Principles

Тактильные поверхности

Интерфейс состоит из «цифровой бумаги». Слои этой «бумаги» размещаются один над другим, и отбрасывают тени. Благодаря этим теням, пользователи лучше понимают основы работы с интерфейсом.



Principles

Графика

- Основополагающие элементы на основе современной типографики, сеток, пространства, масштаба, цвета.
- Графические элементы создают иерархию, значимость и приоритеты. Умышленный выбор цвета – от края до края, образность, крупная типографика, и много белого, создает смелый графический интерфейс, который запоминается и быстро усваивается пользователем.



Principles

Осмысленная анимация

- | От четкости и понятности анимации приложения напрямую зависят затраты времени пользователя на осознание результатов действий, применяемых в данной программе.
- | Анимация — это своеобразная подсказка по управлению интерфейсом Материального дизайна.



Principles

Адаптивный дизайн

Одной из особенностей
Материального дизайна является
его универсальность, то есть
возможность применения трех
выше описанных составляющих на
разных устройствах, таких как
компьютер, телефон, планшет и
др.



Surfaces and shadows

Поверхности и тени.

- Элементы интерфейса должны представлять собой слои, наложенные друг на друга.
- Используйте тени на верхних слоях, чтобы обозначить их иерархию и обратить на них внимание пользователя.



Imagery

Плотность

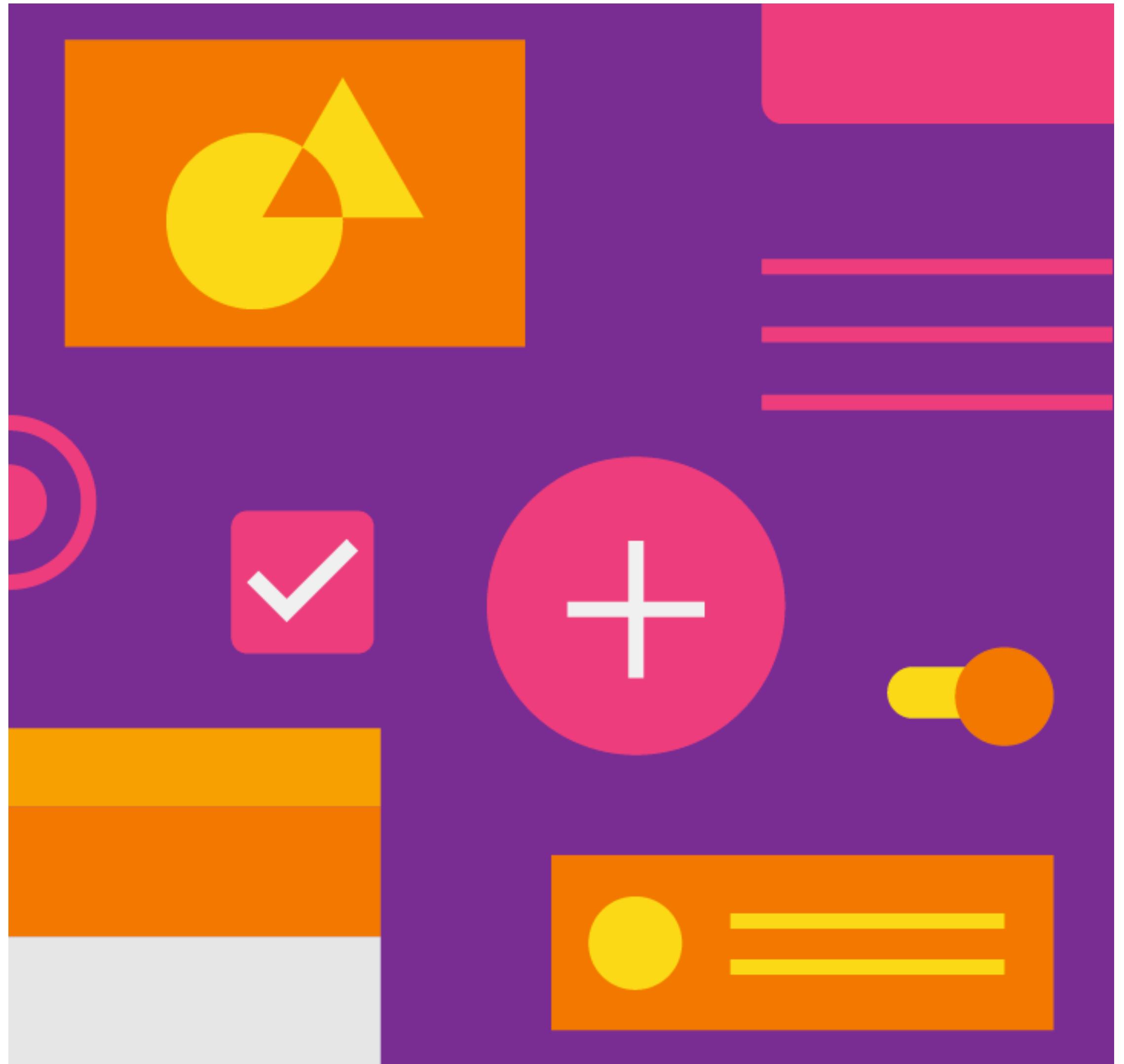
Не оставляйте много свободного пространства между фотографиями и полей по краям экрана. Так вы создадите более наполненный интерфейс.



Colors

Цвета

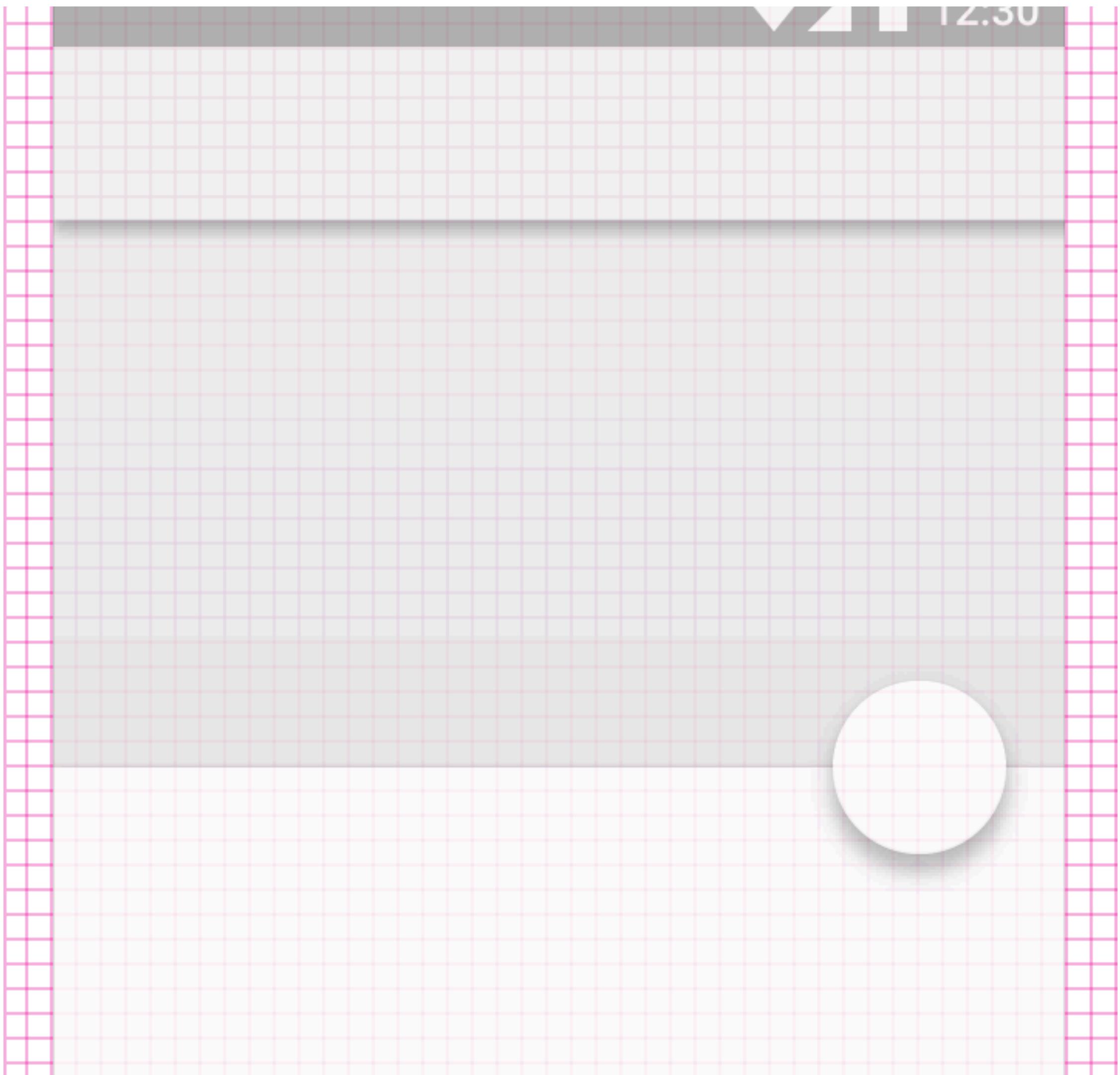
- Используйте контрастные цвета, чтобы сделать акцент на бренде и важных элементах интерфейса.
- Выберите основной и акцентный цвета для всего приложения.



Metrics and keylines

Макет

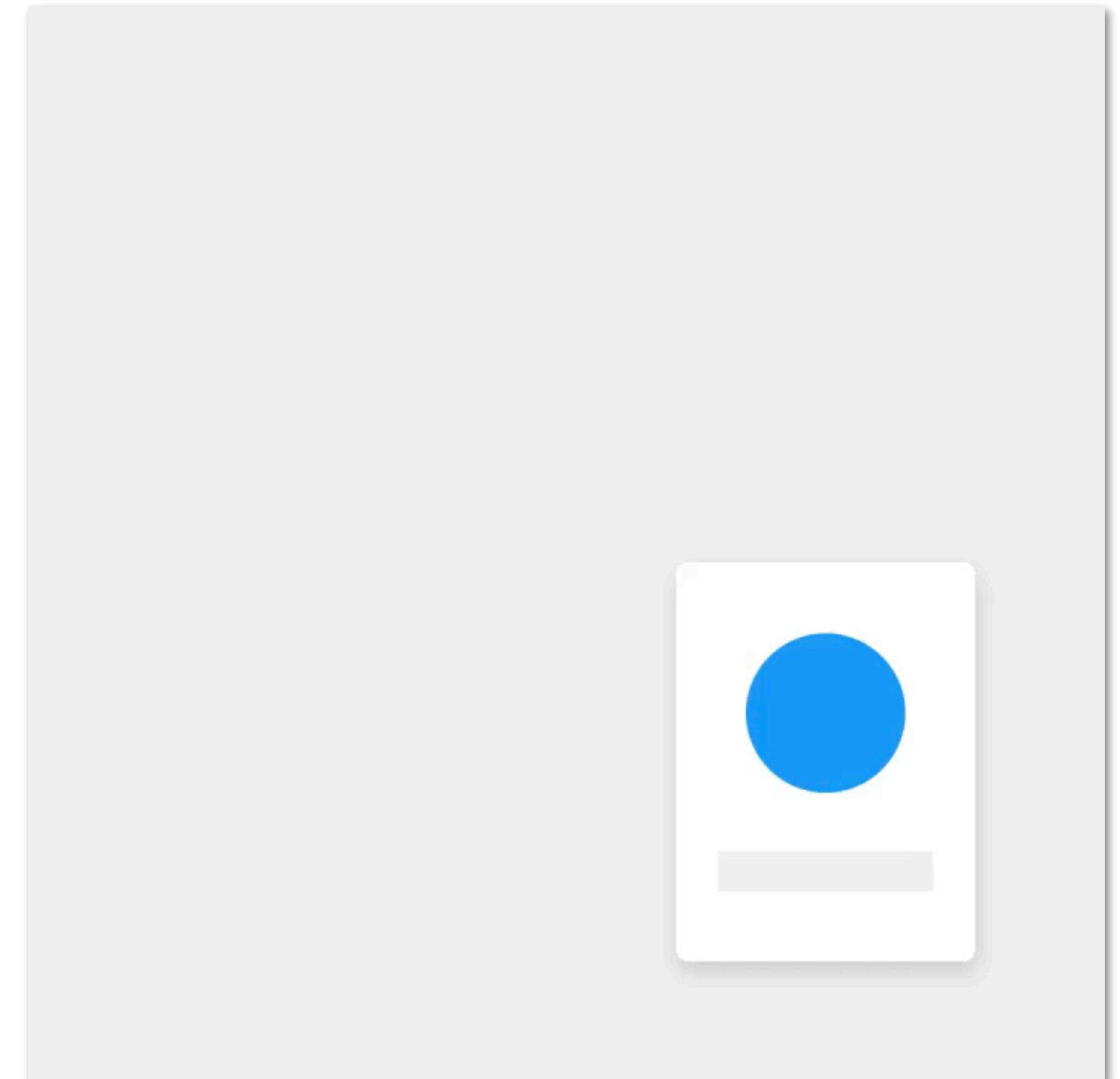
- Макет должен соответствовать стандартной сетке с шагом 8 dp, позволяющей поддерживать единый стиль и удобство чтения.
- Для более точного размещения текста можно использовать сетку с шагом 4 dp.



Meaningful motion

Движущиеся элементы

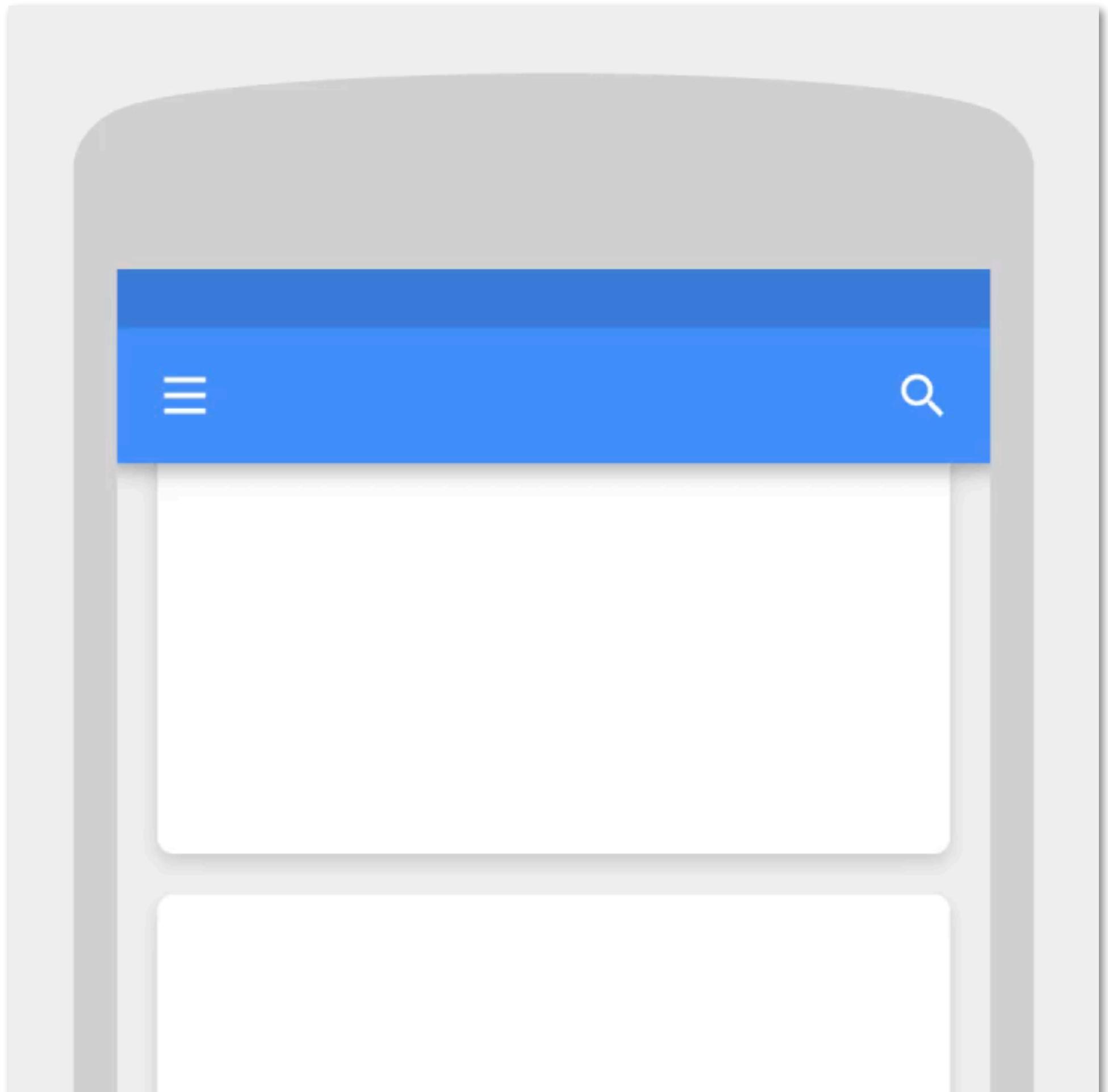
- Движущиеся элементы, например анимация и переходы, должны использоваться для связи одного экрана с другим и не мешать пользователям.
- Анимация делает интерфейс приятным и создает непрерывное визуальное повествование.



Responsive interactions

Реакция на действие

- Основным визуальным механизмом для выражения этой связи является «Рябь от прикосновений».
- Этот прием четко определяет начало и продолжительность события касания, а также его динамические аспекты, такие как сила нажатия при касании.

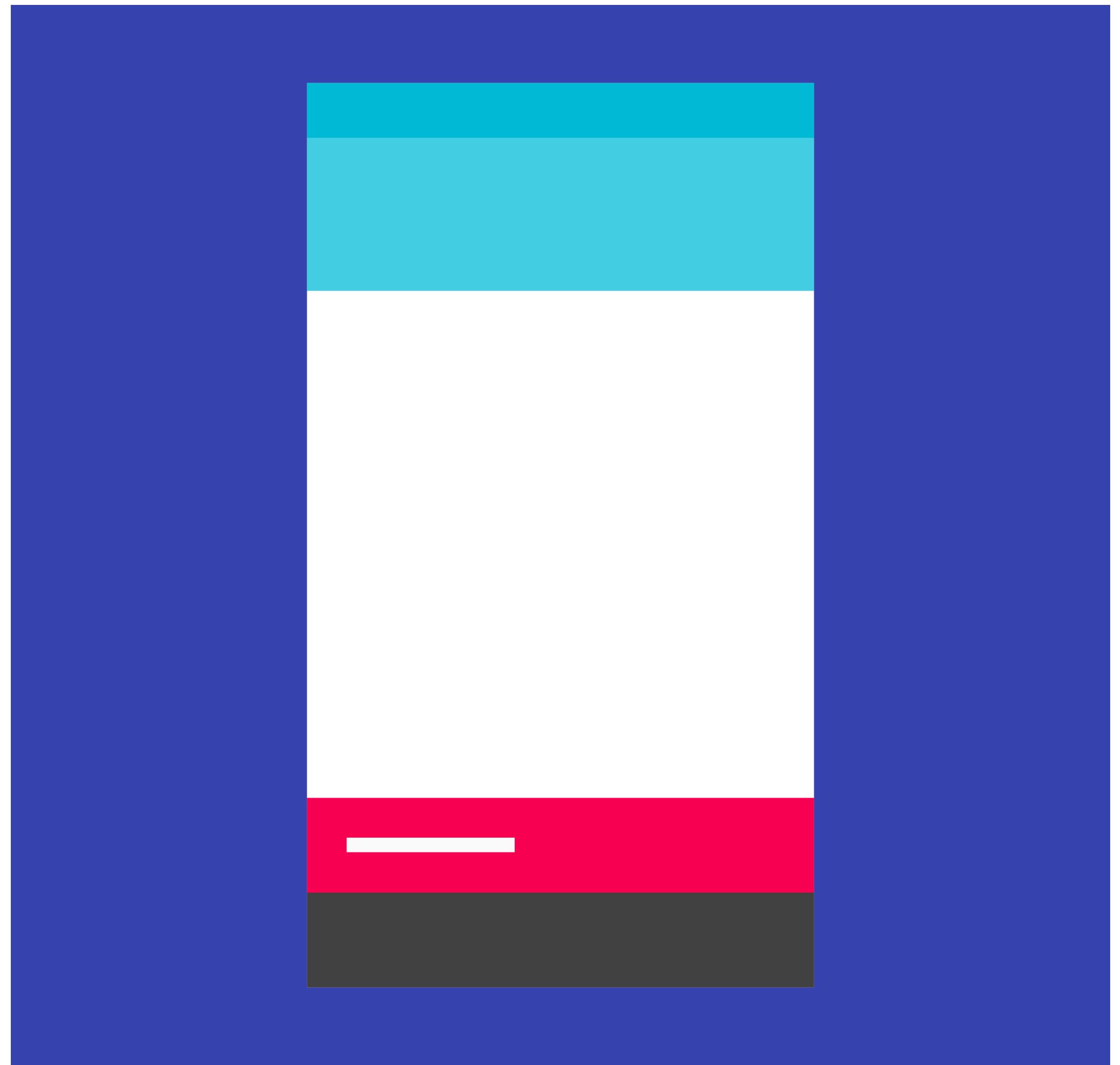


Snackbar



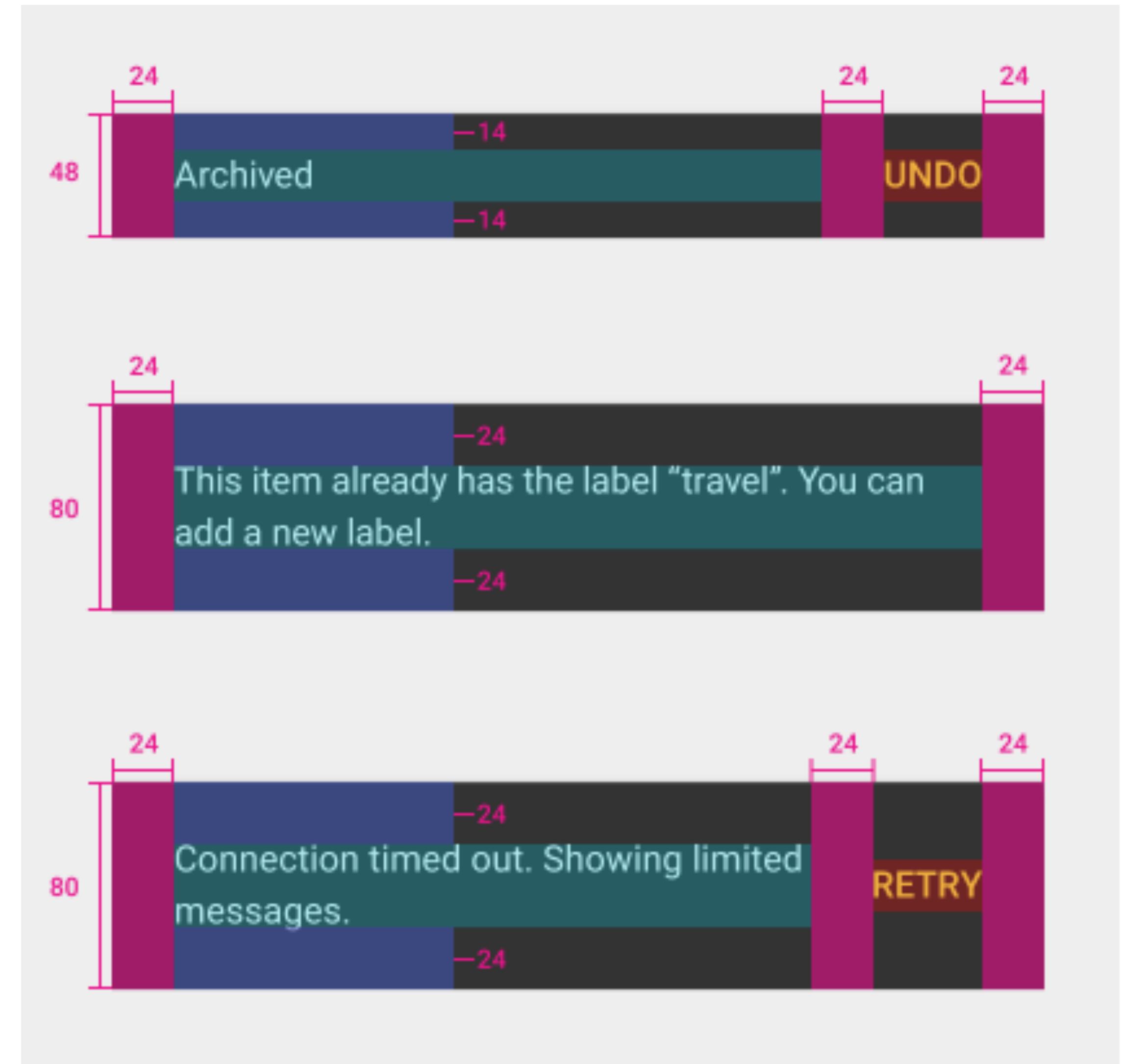
Snackbar

Snackbar – всплывающее сообщение в нижней части экрана, позволяющее обработать одно действие.

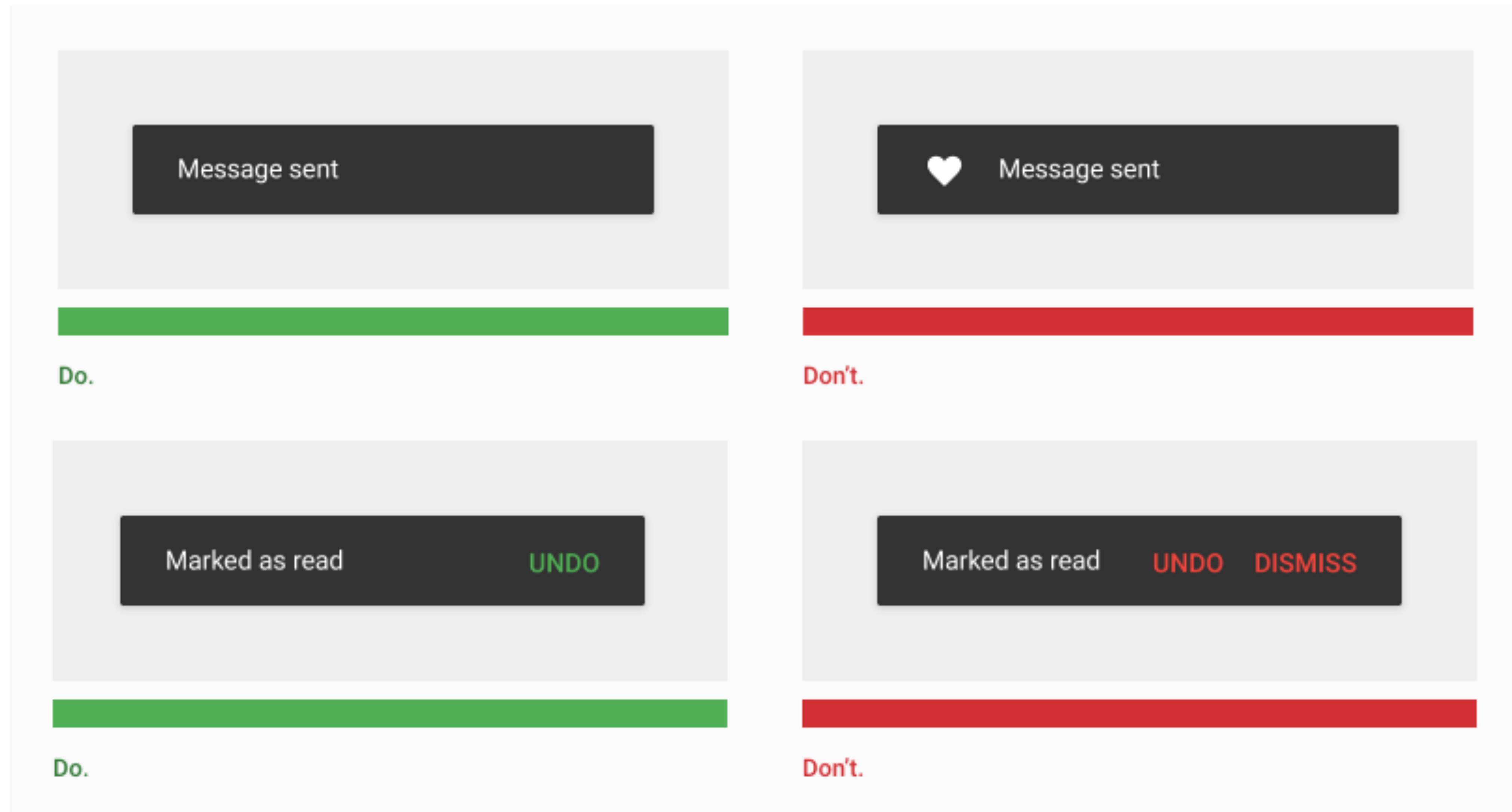


Snackbar

- › Action button: Roboto Medium 14sp, all-caps text
- › Mobile height: 48dp (single-line), 80dp (multi-line)
- › Desktop Snackbar height: 48dp



Snackbar



Snackbar

```
Snackbar.make(view, "Пора кормить кота", Snackbar.LENGTH_LONG)
    .setAction("Покормить") { Log.i("Snackbar", "Кот покормлен") }
    .addCallback(object : Snackbar.Callback() {
        override fun onShown(sb: Snackbar?) {
            Log.i("Snackbar", "Показан")
        }

        override fun onDismissed(transientBottomBar: Snackbar?, event: Int) {
            when(event) {
                DISMISS_EVENT_TIMEOUT -> Log.i("Snackbar", "Закрыт по таймауту")
                DISMISS_EVENT_ACTION -> Log.i("Snackbar", "Закрыт по экшену")
                DISMISS_EVENT_MANUAL -> Log.i("Snackbar", "Закрыт вручную")
                DISMISS_EVENT_SWIPE -> Log.i("Snackbar", "Закрыт свайпом")
            }
        }
    })
    .show()
```

Snackbar

```
Snackbar.make(view, "Пора кормить кота", Snackbar.LENGTH_LONG)
    .setAction("Покормить") { Log.i("Snackbar", "Кот покормлен") }
    .addCallback(object : Snackbar.Callback() {
        override fun onShown(sb: Snackbar?) {
            Log.i("Snackbar", "Показан")
        }

        override fun onDismissed(transientBottomBar: Snackbar?, event: Int) {
            when(event) {
                DISMISS_EVENT_TIMEOUT -> Log.i("Snackbar", "Закрыт по таймауту")
                DISMISS_EVENT_ACTION -> Log.i("Snackbar", "Закрыт по экшену")
                DISMISS_EVENT_MANUAL -> Log.i("Snackbar", "Закрыт вручную")
                DISMISS_EVENT_SWIPE -> Log.i("Snackbar", "Закрыт свайпом")
            }
        }
    })
    .show()
```

Snackbar

```
Snackbar.make(view, "Пора кормить кота", Snackbar.LENGTH_LONG)
    .setAction("Покормить") { Log.i("Snackbar", "Кот покормлен") }
    .addCallback(object : Snackbar.Callback() {
        override fun onShown(sb: Snackbar?) {
            Log.i("Snackbar", "Показан")
        }

        override fun onDismissed(transientBottomBar: Snackbar?, event: Int) {
            when(event) {
                DISMISS_EVENT_TIMEOUT -> Log.i("Snackbar", "Закрыт по таймауту")
                DISMISS_EVENT_ACTION -> Log.i("Snackbar", "Закрыт по экшену")
                DISMISS_EVENT_MANUAL -> Log.i("Snackbar", "Закрыт вручную")
                DISMISS_EVENT_SWIPE -> Log.i("Snackbar", "Закрыт свайпом")
            }
        }
    })
    .show()
```

Snackbar

```
Snackbar.make(view, "Пора кормить кота", Snackbar.LENGTH_LONG)
    .setAction("Покормить") { Log.i("Snackbar", "Кот покормлен") }
    .addCallback(object : Snackbar.Callback() {
        override fun onShown(sb: Snackbar?) {
            Log.i("Snackbar", "Показан")
        }

        override fun onDismissed(transientBottomBar: Snackbar?, event: Int) {
            when(event) {
                DISMISS_EVENT_TIMEOUT -> Log.i("Snackbar", "Закрыт по таймауту")
                DISMISS_EVENT_ACTION -> Log.i("Snackbar", "Закрыт по экшену")
                DISMISS_EVENT_MANUAL -> Log.i("Snackbar", "Закрыт вручную")
                DISMISS_EVENT_SWIPE -> Log.i("Snackbar", "Закрыт свайпом")
            }
        }
    })
    .show()
```

Snackbar

```
Snackbar.make(view, "Пора кормить кота", Snackbar.LENGTH_LONG)
    .setAction("Покормить") { Log.i("Snackbar", "Кот покормлен") }
    .addCallback(object : Snackbar.Callback() {
        override fun onShown(sb: Snackbar?) {
            Log.i("Snackbar", "Показан")
        }

        override fun onDismissed(transientBottomBar: Snackbar?, event: Int) {
            when(event) {
                DISMISS_EVENT_TIMEOUT -> Log.i("Snackbar", "Закрыт по таймауту")
                DISMISS_EVENT_ACTION -> Log.i("Snackbar", "Закрыт по экшену")
                DISMISS_EVENT_MANUAL -> Log.i("Snackbar", "Закрыт вручную")
                DISMISS_EVENT_SWIPE -> Log.i("Snackbar", "Закрыт свайпом")
            }
        }
    })
    .show()
```

Snackbar

```
Snackbar.make(view, "Пора кормить кота", Snackbar.LENGTH_LONG)
    .setAction("Покормить") { Log.i("Snackbar", "Кот покормлен") }
    .addCallback(object : Snackbar.Callback() {
        override fun onShown(sb: Snackbar?) {
            Log.i("Snackbar", "Показан")
        }

        override fun onDismissed(transientBottomBar: Snackbar?, event: Int) {
            when(event) {
                DISMISS_EVENT_TIMEOUT -> Log.i("Snackbar", "Закрыт по таймауту")
                DISMISS_EVENT_ACTION -> Log.i("Snackbar", "Закрыт по экшену")
                DISMISS_EVENT_MANUAL -> Log.i("Snackbar", "Закрыт вручную")
                DISMISS_EVENT_SWIPE -> Log.i("Snackbar", "Закрыт свайпом")
            }
        }
    })
    .show()
```

Navigation Component



Navigation Component

| Navigation Component - инструмент, предназначенный облегчить реализацию навигации внутри приложения.

Navigation Component

- | Navigation Component - инструмент, предназначенный облегчить реализацию навигации внутри приложения.
- | Способен обрабатывать переходы между экранами по различным событиям и корректные возвраты назад.

Navigation Component

- | Navigation Component - инструмент, предназначенный облегчить реализацию навигации внутри приложения.
- | Способен обрабатывать переходы между экранами по различным событиям и корректные возвраты назад.
- | Включает в себя:
 - > Navigation graph
 - > NavHost
 - > NavController

Navigation Component

| Navigation graph - XML файл с описанием навигации между компонентами приложения, называемыми назначениями (destinations).

Navigation Component

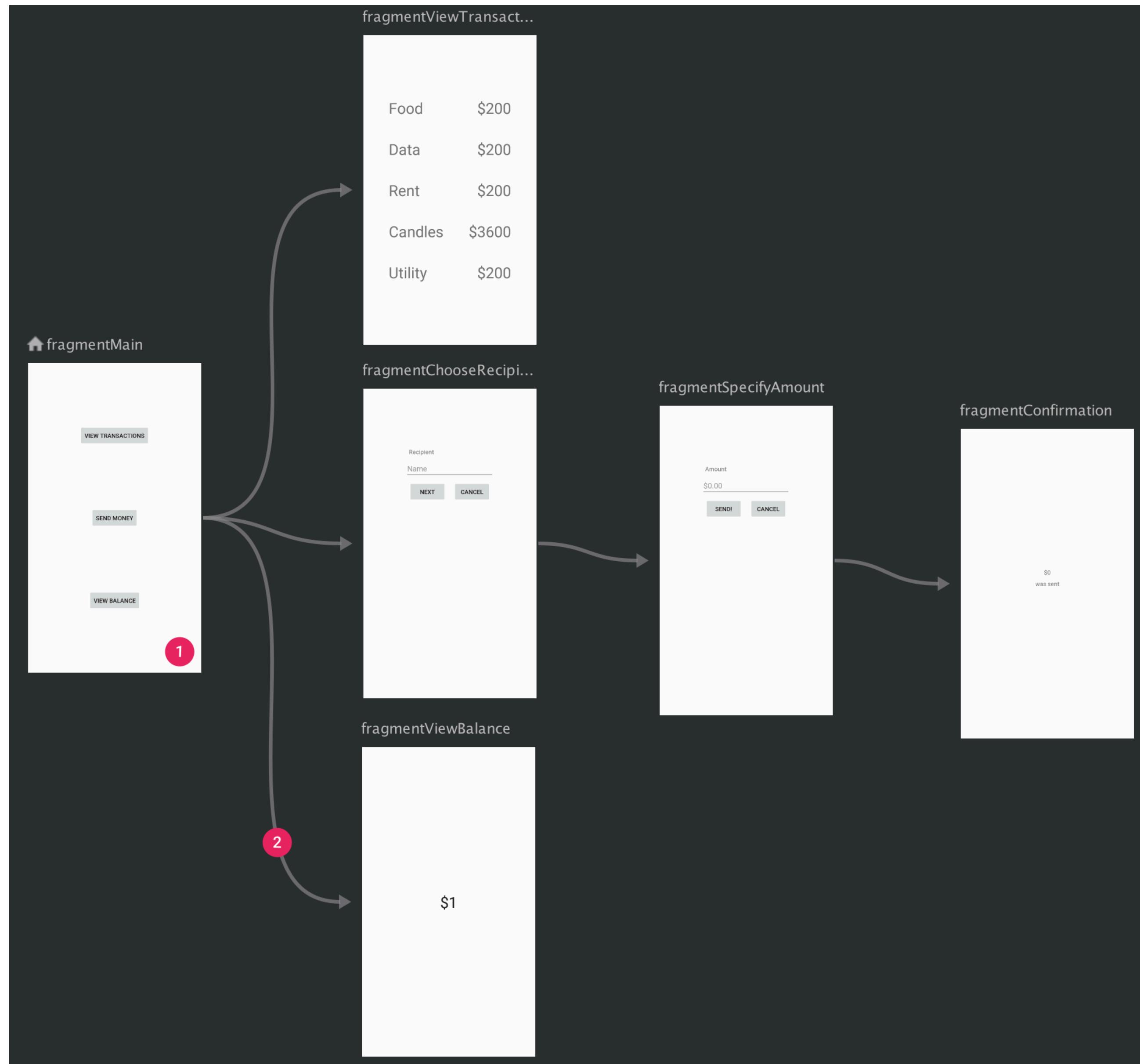
| Navigation graph - XML файл с описанием навигации между компонентами приложения, называемыми назначениями (destinations).

```
<?xml version="1.0" encoding="utf-8"?>
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/mobile_navigation"
    app:startDestination="@+id/nav_home">

    <fragment
        android:id="@+id/nav_home"
        android:name="com.example.uicomponents.ui.home.HomeFragment"
        android:label="@string/menu_home"
        tools:layout="@layout/fragment_home"/>

    <fragment
        android:id="@+id/nav_gallery"
        android:name="com.example.uicomponents.ui.gallery.GalleryFragment"
        android:label="@string/menu_gallery"
        tools:layout="@layout/fragment_gallery" />
</navigation>
```

Navigation Component



Navigation Component

| NavHost - пустой контейнер для отображения в нём назначений.

Navigation Component

- | NavHost - пустой контейнер для отображения в нём назначений.
- | Дефолтная реализация - NavHostFragment.

Navigation Component

- | NavHost - пустой контейнер для отображения в нём назначений.
- | Дeфолтная реализация - NavHostFragment.

```
<androidx.constraintlayout.widget.ConstraintLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent">  
  
<fragment  
    android:id="@+id/nav_host_fragment"  
    android:name="androidx.navigation.fragment.NavHostFragment"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    app:defaultNavHost="true"  
    app:layout_constraintLeft_toLeftOf="parent"  
    app:layout_constraintRight_toRightOf="parent"  
    app:layout_constraintTop_toTopOf="parent"  
    app:navGraph="@navigation/mobile_navigation" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Navigation Component

| NavController - объект, управляющий навигацией для конкретного NavHost.

Navigation Component

- | NavController - объект, управляющий навигацией для конкретного NavHost.
- | Для осуществления навигации необходимо сообщить NavController путь в навигационном графе или конкретное назначение.

Navigation Component

- | NavController - объект, управляющий навигацией для конкретного NavHost.
- | Для осуществления навигации необходимо сообщить NavController путь в навигационном графе или конкретное назначение.
- | Нужное назначение будет отображено внутри NavHost.

Navigation Drawer



Navigation Drawer

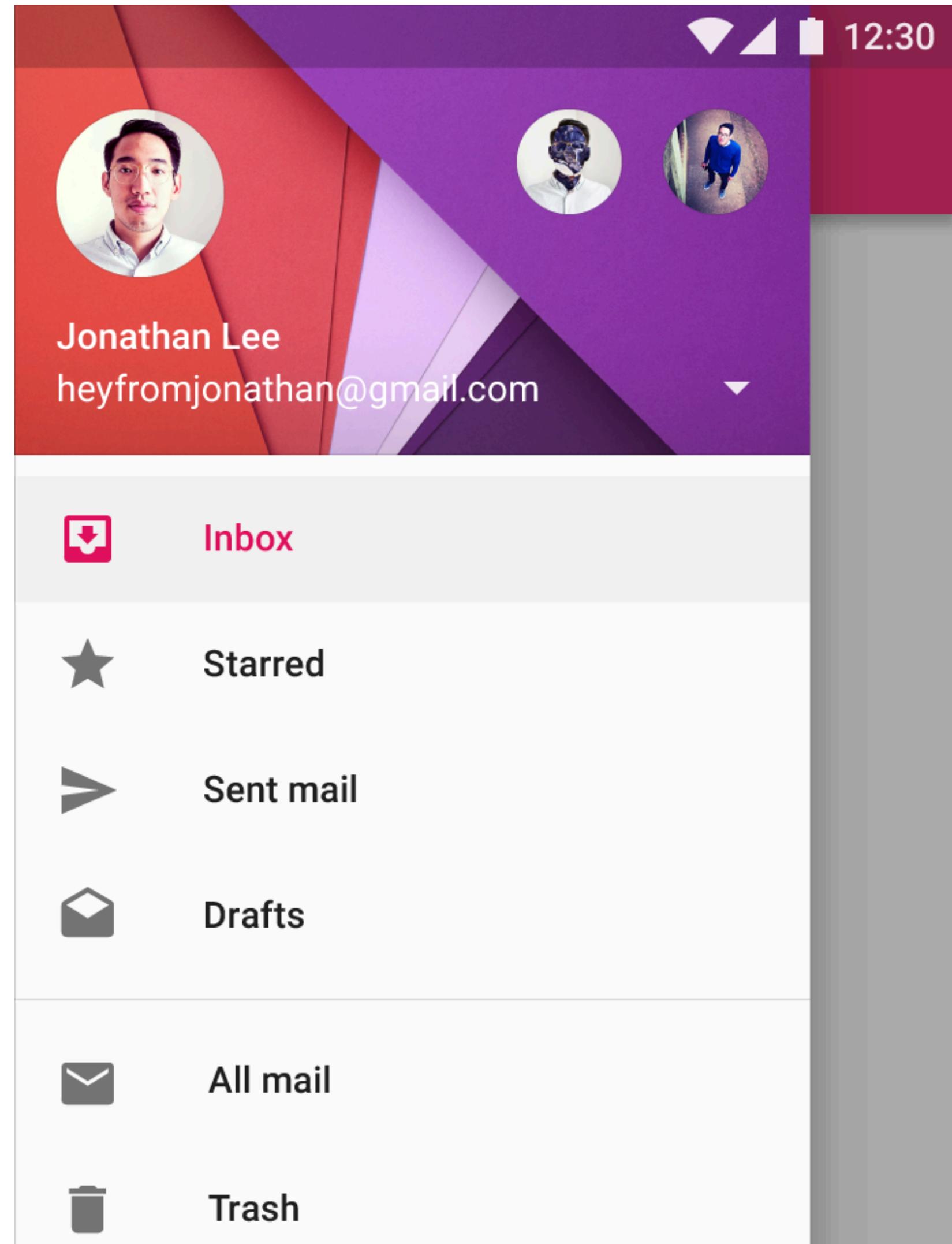
Navigation Drawer – элемент пользовательского интерфейса, предназначенный для навигации по основным компонентам Android приложения.



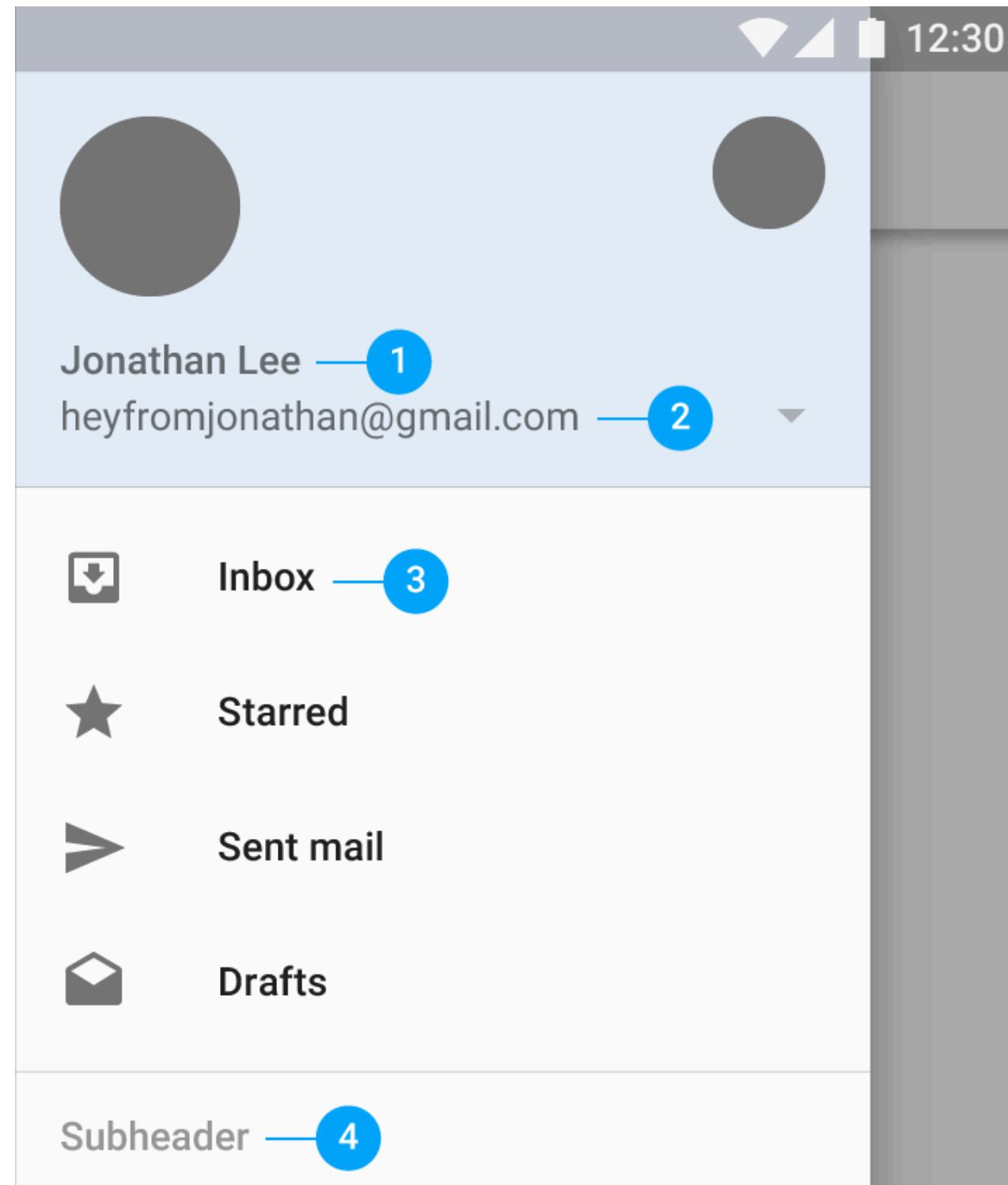
Navigation Drawer

Размещается по всей высоте экрана, перекрывая основной экран приложения, но находясь под строкой состояния.

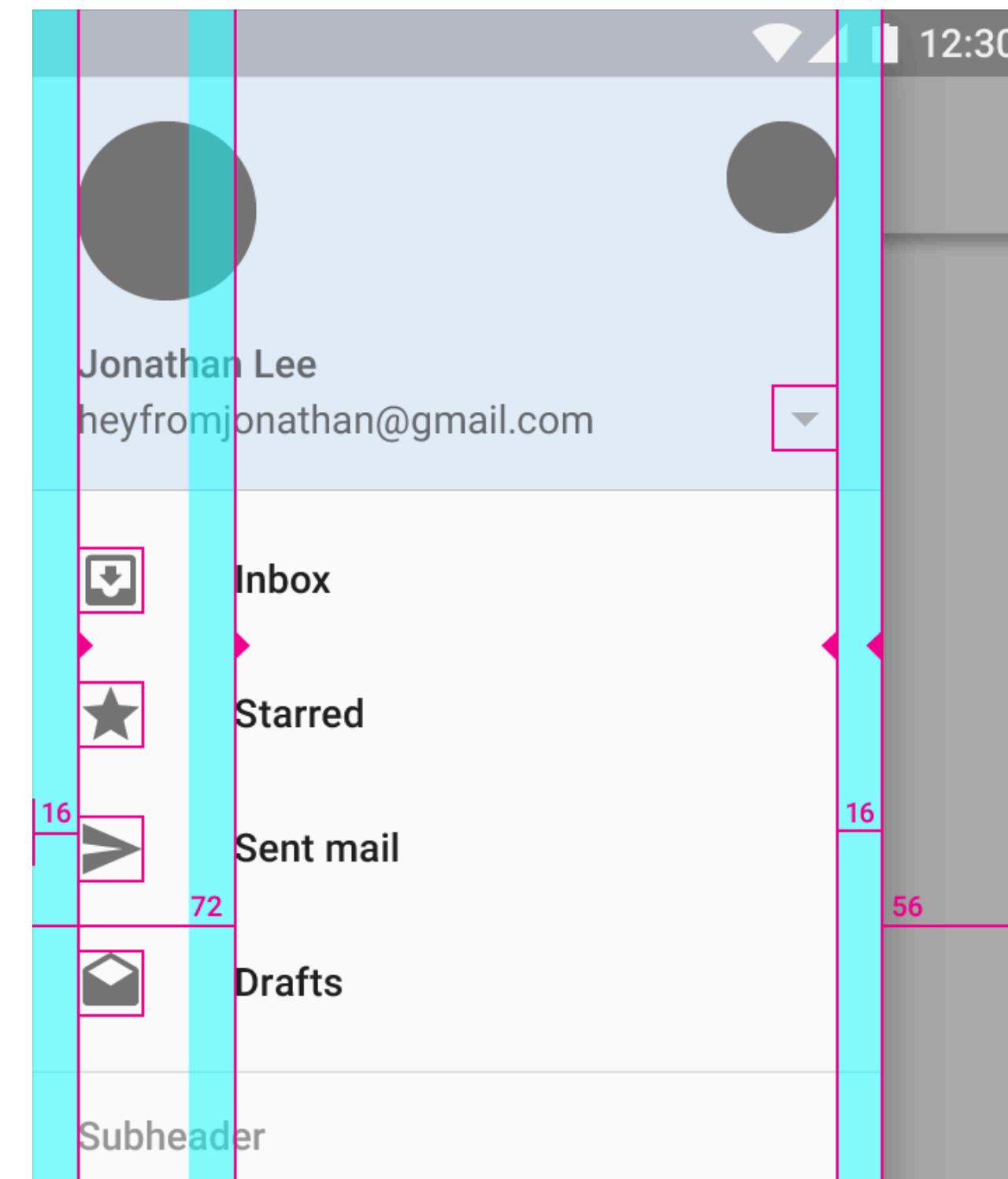
Состоит из хидера и панели навигационного меню.



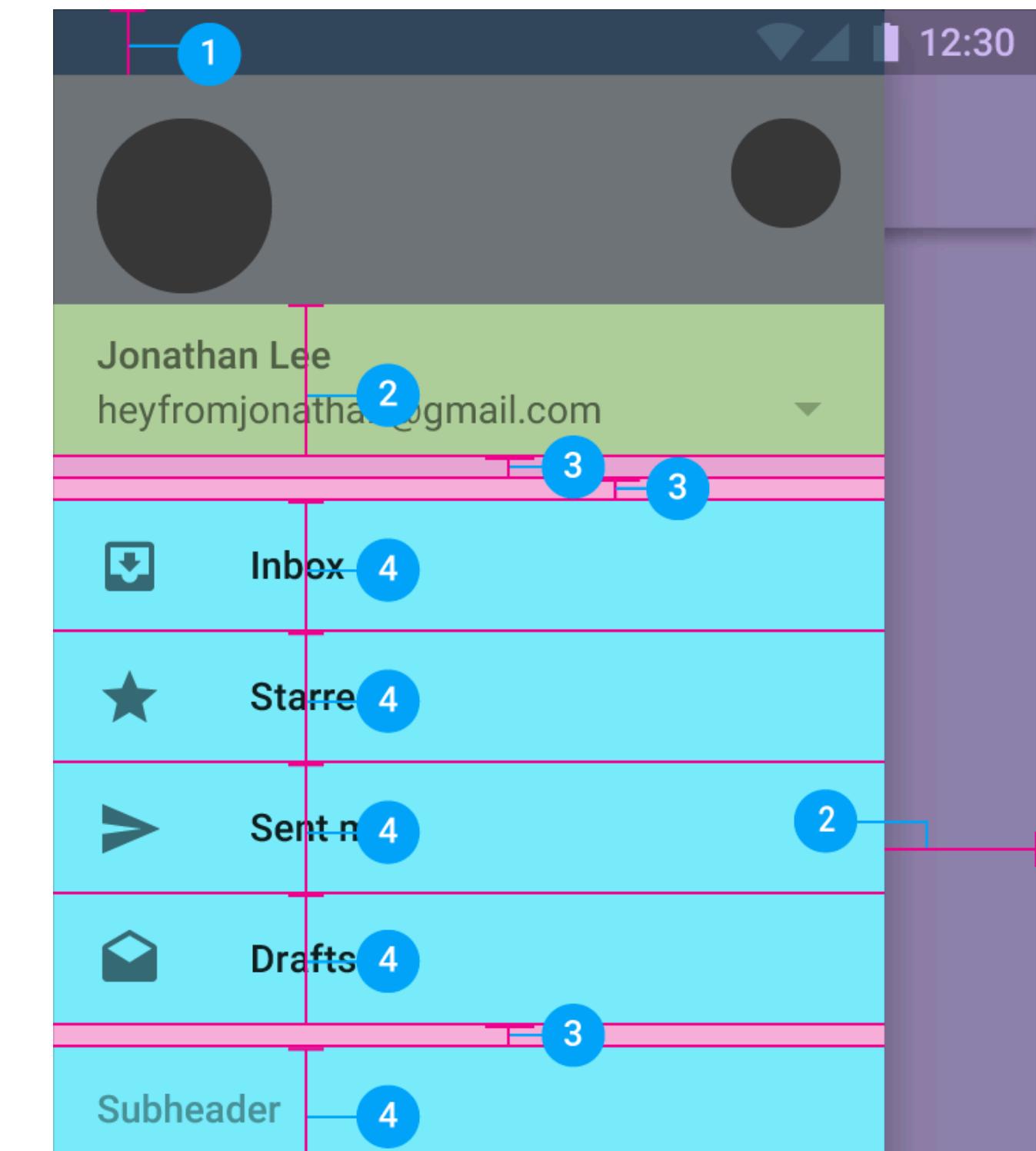
Navigation Drawer



1. Name: Roboto Medium, 14sp, #FFFFFF
2. Email address: Roboto Regular, 14sp, #FFFFFF
3. List item: Roboto Medium, 14sp, 87% #000000
4. Subheader: Roboto Medium, 14sp, 54% #000000. It aligns to the 16dp keyline.



- › Icons align at screen left and right margins: 16dp
- › Icon values: 54% #000000
- › Content associated with an icon or avatar left margin: 72dp



1. Status bar: 24dp
2. Subtitle: 56dp
3. Space between content areas: 8dp
4. Subtitles and list items: 48dp

Navigation Drawer

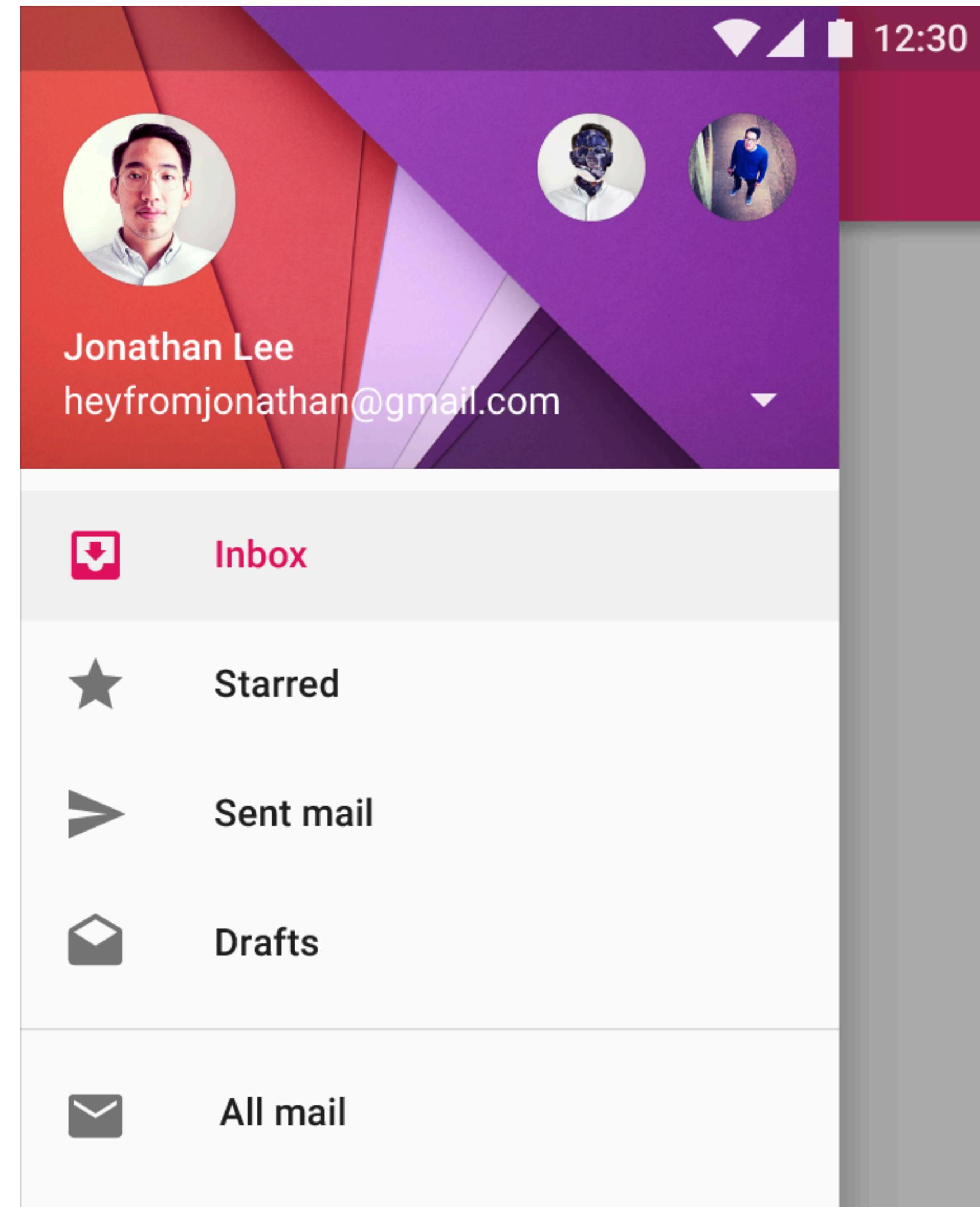
```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:showIn="navigation_view">

    <group android:checkableBehavior="single">

        <item
            android:id="@+id/nav_home"
            android:icon="@drawable/ic_menu_camera"
            android:title="@string/menu_home" />

        <item
            android:id="@+id/nav_gallery"
            android:icon="@drawable/ic_menu_gallery"
            android:title="@string/menu_gallery" />

        <item
            android:id="@+id/nav_slideshow"
            android:icon="@drawable/ic_menu_slideshow"
            android:title="@string/menu_slideshow" />
    </group>
</menu>
```



AppBar scrollable regions

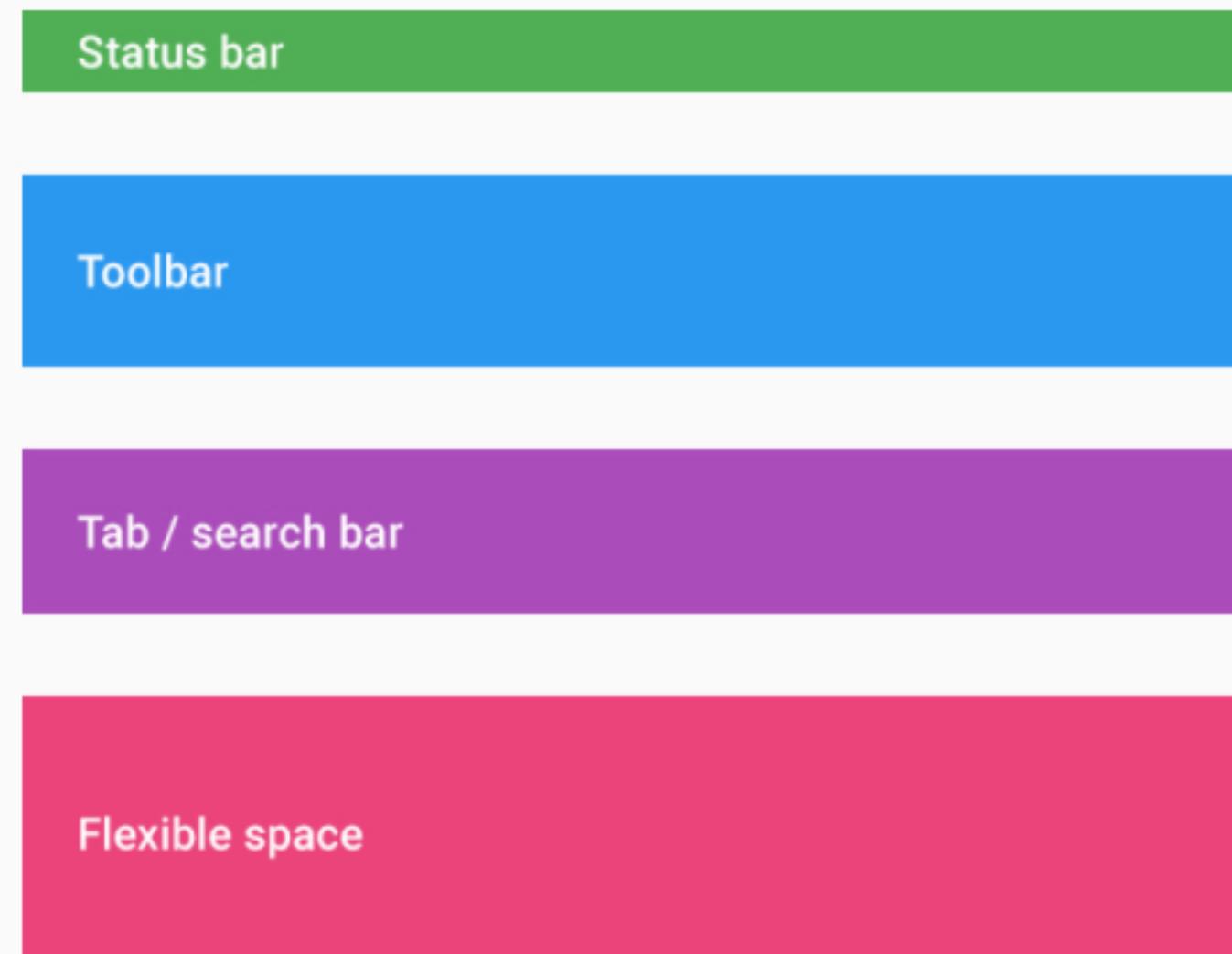


App bar scrollable regions

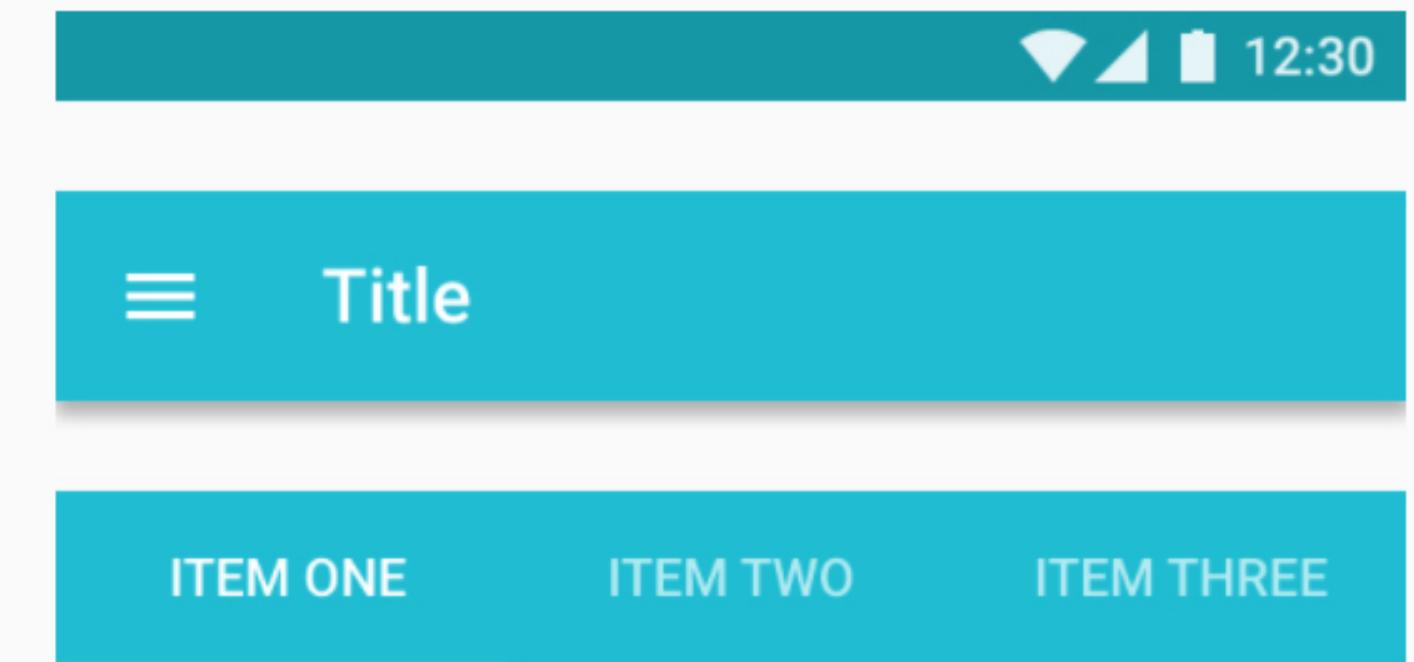
App bar scrollable regions

When designing scrolling behavior, app bars contain four main regions (referred to as **blocks**) that comprise the scrolling structure:

- Status bar
- Tool bar
- Tab bar/search bar
- Flexible space: space to accommodate a desired aspect ratio for images or extended app bars



App bar components can include: status bar, navigation bar, tab/search bar, and flexible space



Large title

Example of a status bar, navigation bar, tab/search bar, and flexible space

Standard App Bar

Standard app bar

Specs:

The standard app bar height is 56dp on mobile and 64dp on larger screen sizes.

The app bar has two scrolling options:

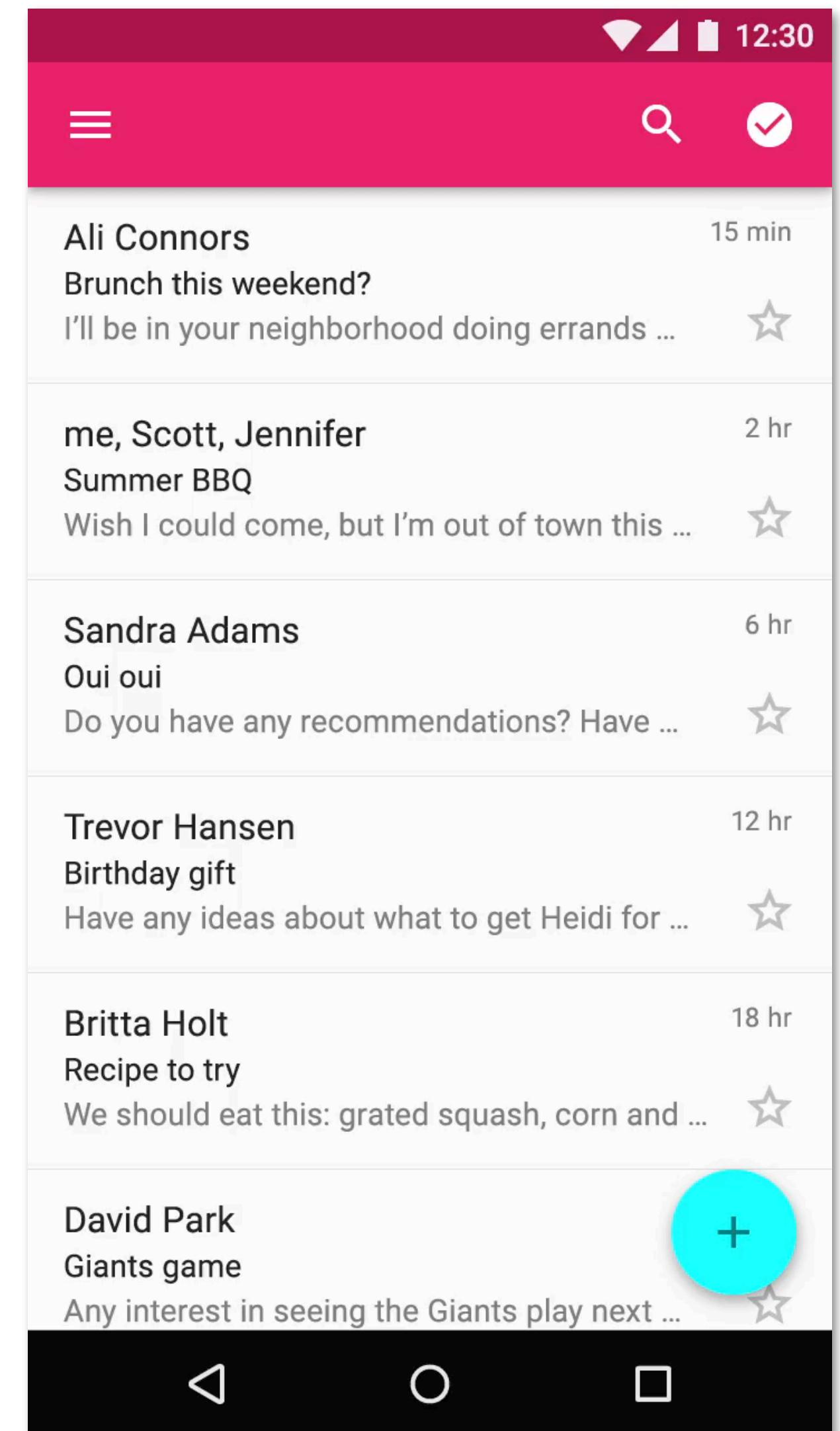
- The app bar can **scroll off-screen** with the content and return when the user reverse scrolls
- The app bar can **stay fixed at the top** with content scrolling under it

Status bar	24dp
Toolbar	56dp / 64dp

The standard app bar

Status bar height: 24dp

Toolbar height: 56dp / 64dp



App Bar with tabs

App bar with tabs

The **standard app bar** component that can include the following blocks: a toolbar, tab bar, or flexible space.

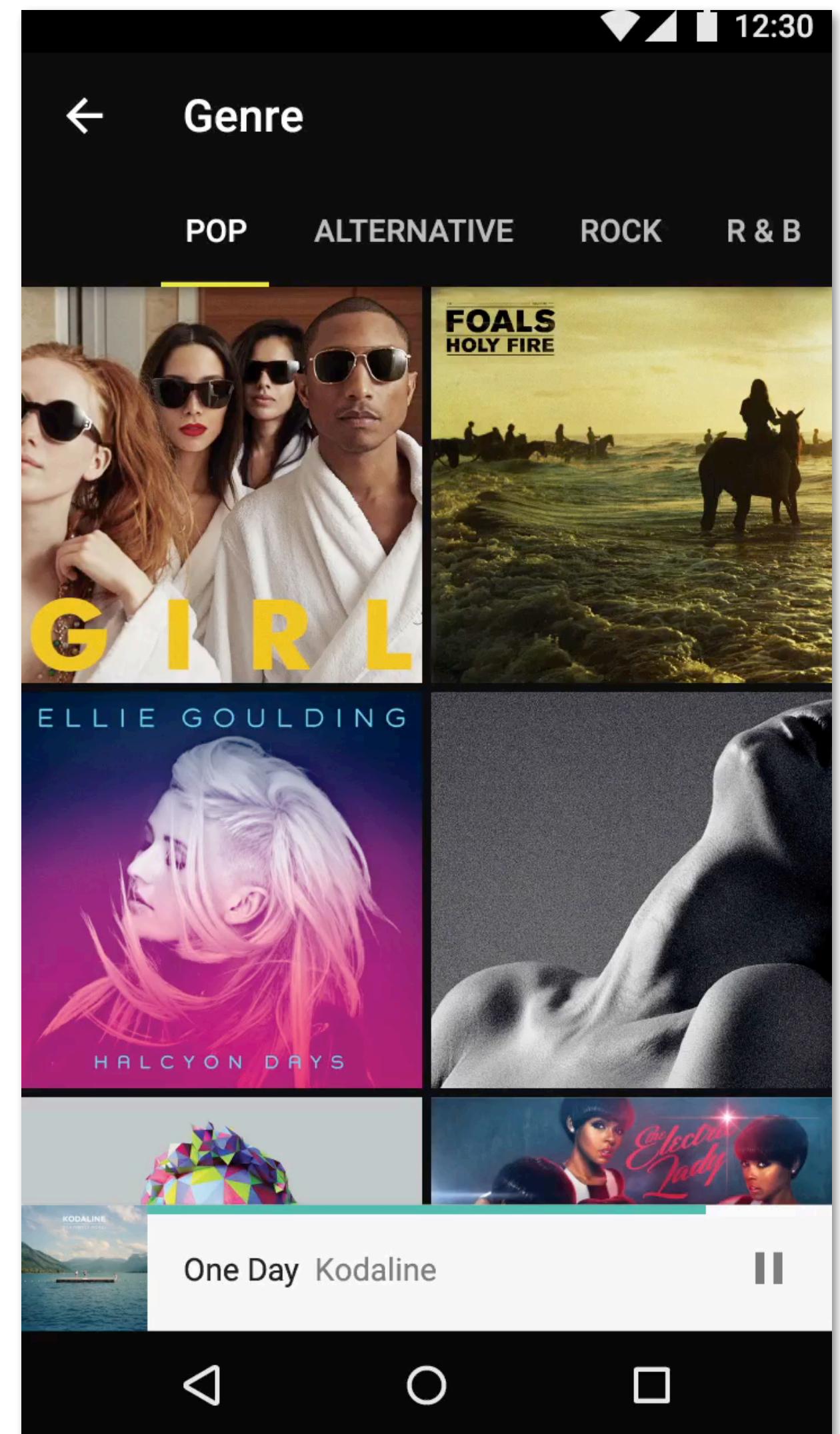
Tabs may have one of these behaviors:

- The tab bar stays **anchored at the top**, while the toolbar scrolls off
- The app bar stays anchored at the top, with the **content scrolling underneath**
- Both the toolbar and tab bar **scroll off with content**. The tab bar returns on reverse-scroll, and the toolbar returns on complete reverse scroll

Status bar	24dp
Toolbar	56dp / 64dp
Tab bar	48dp

Status bar, toolbar, and tab bar

Status bar height: 24dp
Toolbar height: 56dp / 64dp
Tab bar height: 48dp



App Bar with flexible space

Flexible space

Because the app bar is flexible, it can be extended to accommodate larger typography or images. To extend the app bar, add a flexible space block.

Flexible space may be displayed one of two ways:

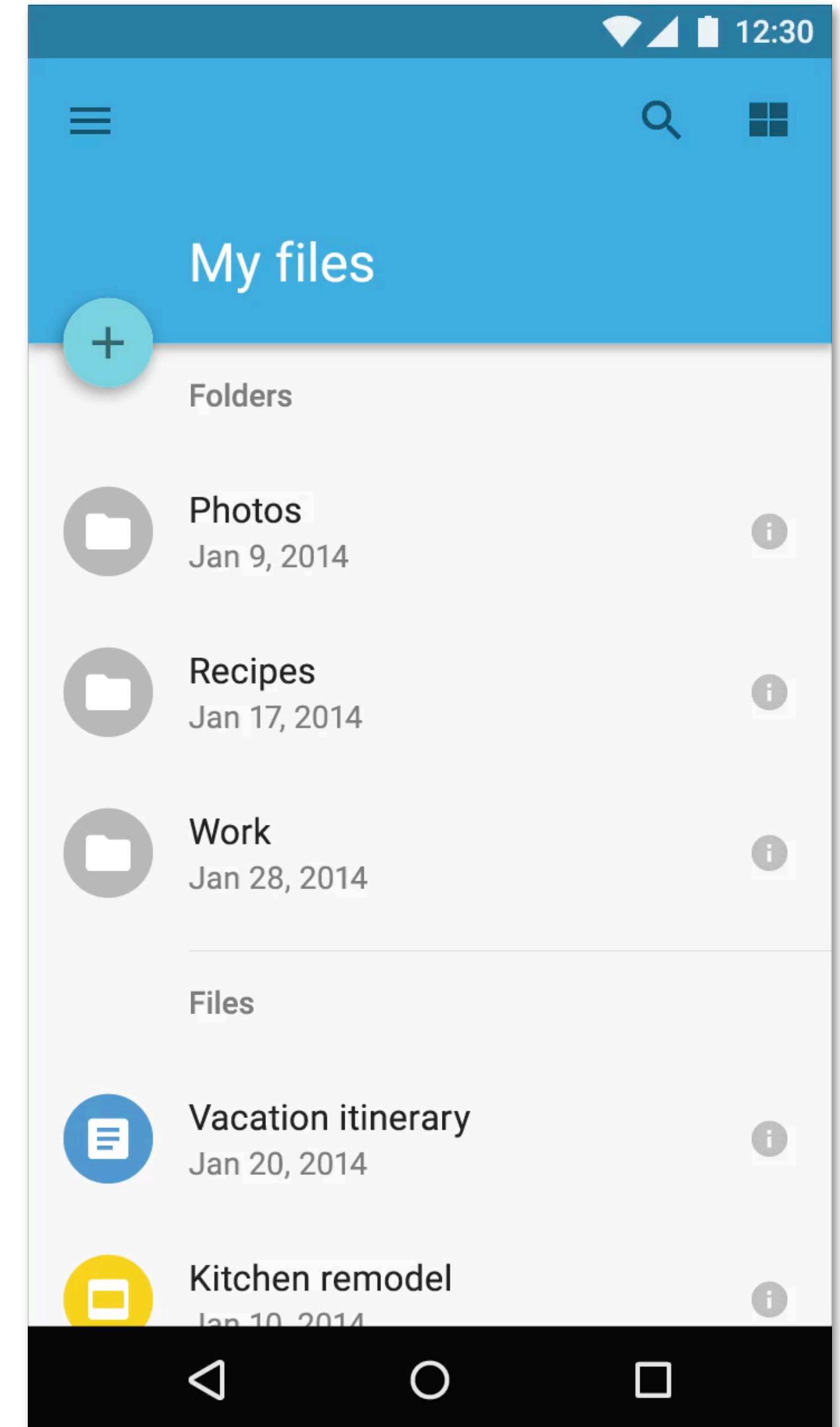
- The **flexible space shrinks** until only the toolbar remains. The title shrinks to 20sp in the navigation bar. When scrolling to the top of the page, the flexible space and the title grow into place again.
- The **whole app bar scrolls off**. When the user reverse scrolls, the toolbar returns anchored to the top. When scrolling all the way back, the flexible space and the title grow into place again.



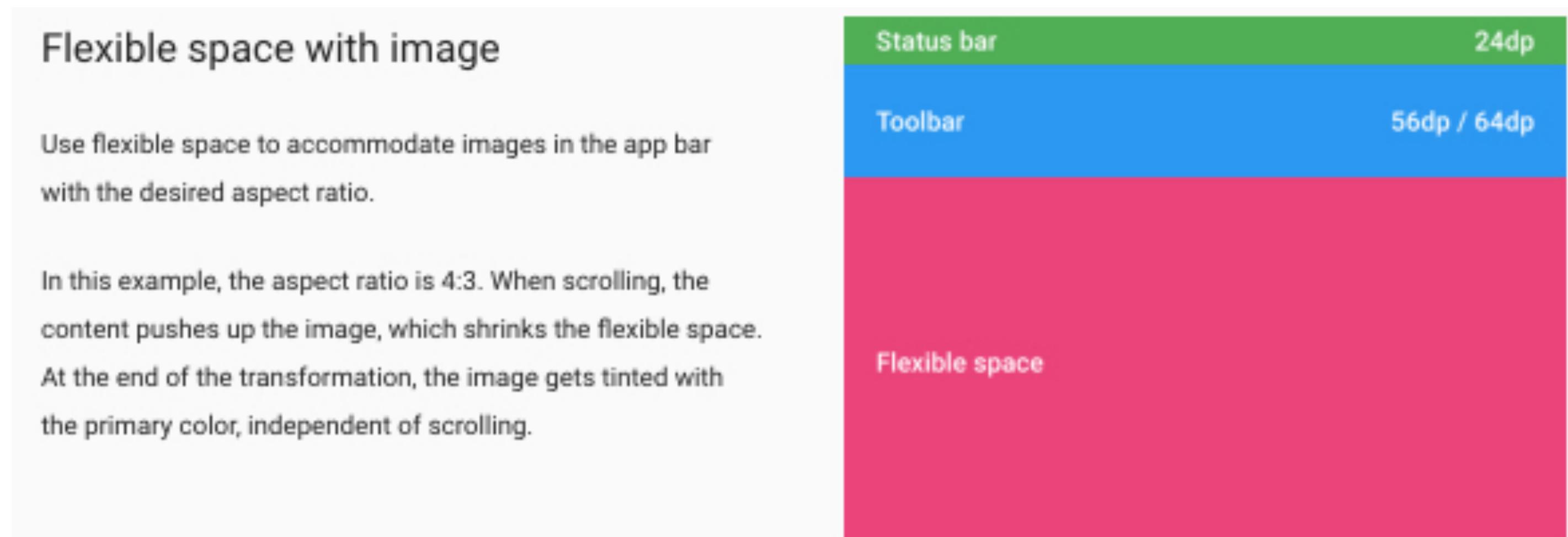
Status bar, toolbar, and flexible space

Status bar height: 24dp

Toolbar height: 56dp / 64dp



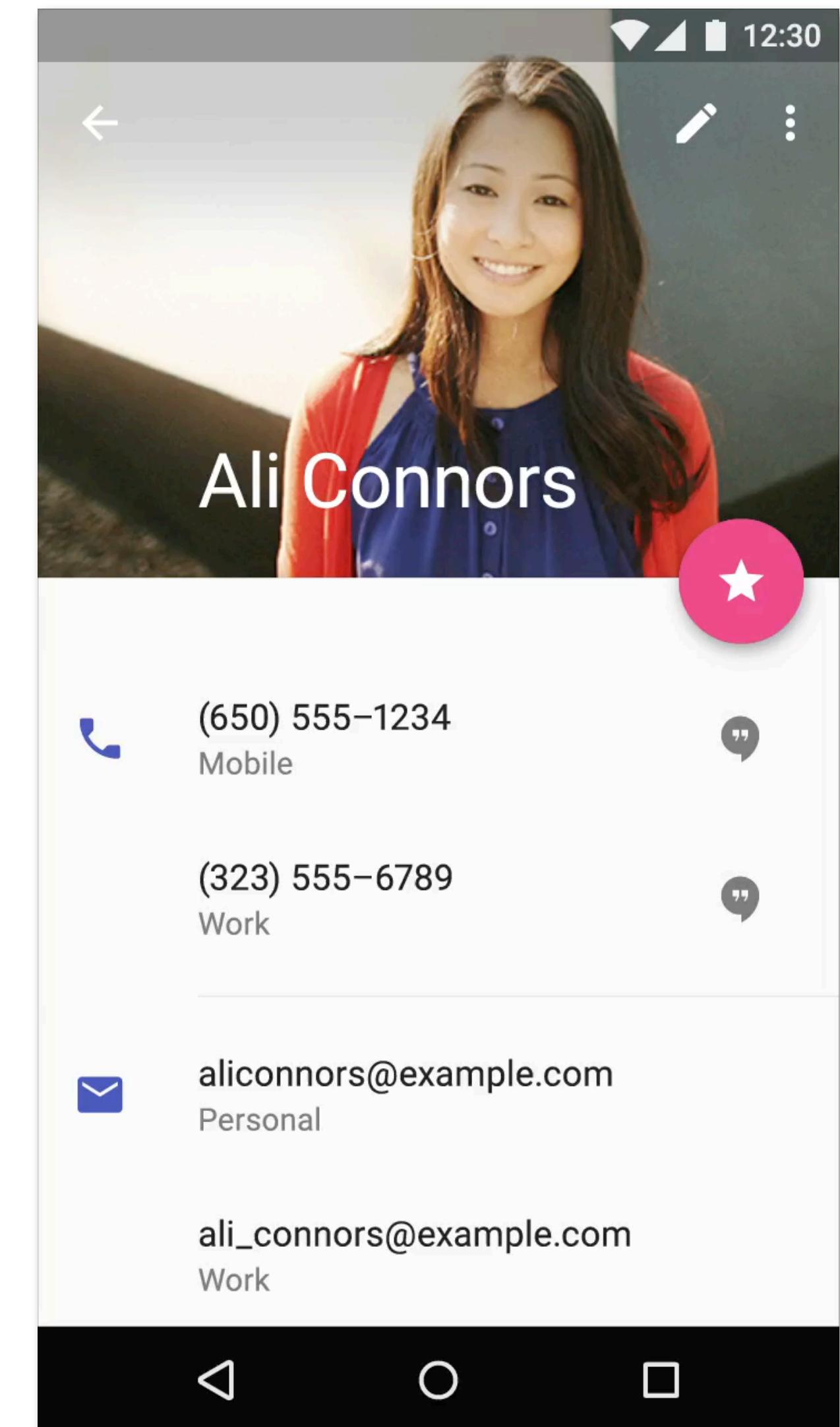
App Bar with flexible space & image



Status bar, toolbar, and flexible space

Status bar height: 24dp

Toolbar height: 56dp / 64dp



CoordinatorLayout

CoordinatorLayout – продвинутый FrameLayout, позволяющий координировать некоторые зависимости между включенными в него виджетами.

```
<androidx.coordinatorlayout.widget.CoordinatorLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:fitsSystemWindows="true">  
  
<com.google.android.material.appbar.AppBarLayout  
    android:id="@+id/app_bar"  
    android:layout_width="match_parent"  
    android:layout_height="@dimen/app_bar_height"  
    android:fitsSystemWindows="true"  
    android:theme="@style/AppTheme.AppBarOverlay">  
  
<com.google.android.material.appbar.CollapsingToolbarLayout  
    android:id="@+id/toolbar_layout"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:fitsSystemWindows="true"  
    app:contentScrim="?attr/colorPrimary"  
    app:layout_scrollFlags="scroll|exitUntilCollapsed"  
    app:toolbarId="@+id/toolbar">  
  
<androidx.appcompat.widget.Toolbar  
    android:id="@+id/toolbar"  
    android:layout_width="match_parent"  
    android:layout_height="?attr/actionBarSize"  
    app:layout_collapseMode="pin"  
    app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
    </com.google.android.material.appbar.CollapsingToolbarLayout>  
</com.google.android.material.appbar.AppBarLayout>  
  
<include layout="@layout/content_scrolling" />  
  
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

CoordinatorLayout.Behavior

CoordinatorLayout.Behavior позволяет определить свою реализацию для компонентов внутри CoordinatorLayout.

- › YourBehaviour: CoordinatorLayout.Behavior<ChildView>
- › override fun layoutDependsOn(parent: CoordinatorLayout, child: V, dependency: View): Boolean
- › override fun onDependentViewChanged(parent: CoordinatorLayout, child: V, dependency: View): Boolean

AppBarLayout

AppBarLayout - вертикальный LinearLayout, обеспечивающий поведение своих дочерних элементов во время скроллинга. Поведение дочерних элементов определяется через атрибут app:layout_scrollFlags.

```
<androidx.coordinatorlayout.widget.CoordinatorLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:fitsSystemWindows="true">  
  
    <com.google.android.material.appbar.AppBarLayout  
        android:id="@+id/app_bar"  
        android:layout_width="match_parent"  
        android:layout_height="@dimen/app_bar_height"  
        android:fitsSystemWindows="true"  
        android:theme="@style/AppTheme.AppBarOverlay">  
  
        <com.google.android.material.appbar.CollapsingToolbarLayout  
            android:id="@+id/toolbar_layout"  
            android:layout_width="match_parent"  
            android:layout_height="match_parent"  
            android:fitsSystemWindows="true"  
            app:contentScrim="?attr/colorPrimary"  
            app:layout_scrollFlags="scroll|exitUntilCollapsed"  
            app:toolbarId="@+id/toolbar">  
  
            <androidx.appcompat.widget.Toolbar  
                android:id="@+id/toolbar"  
                android:layout_width="match_parent"  
                android:layout_height="?attr/actionBarSize"  
                app:layout_collapseMode="pin"  
                app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
        </com.google.android.material.appbar.CollapsingToolbarLayout>  
    </com.google.android.material.appbar.AppBarLayout>  
  
    <include layout="@layout/content_scrolling" />  
  
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

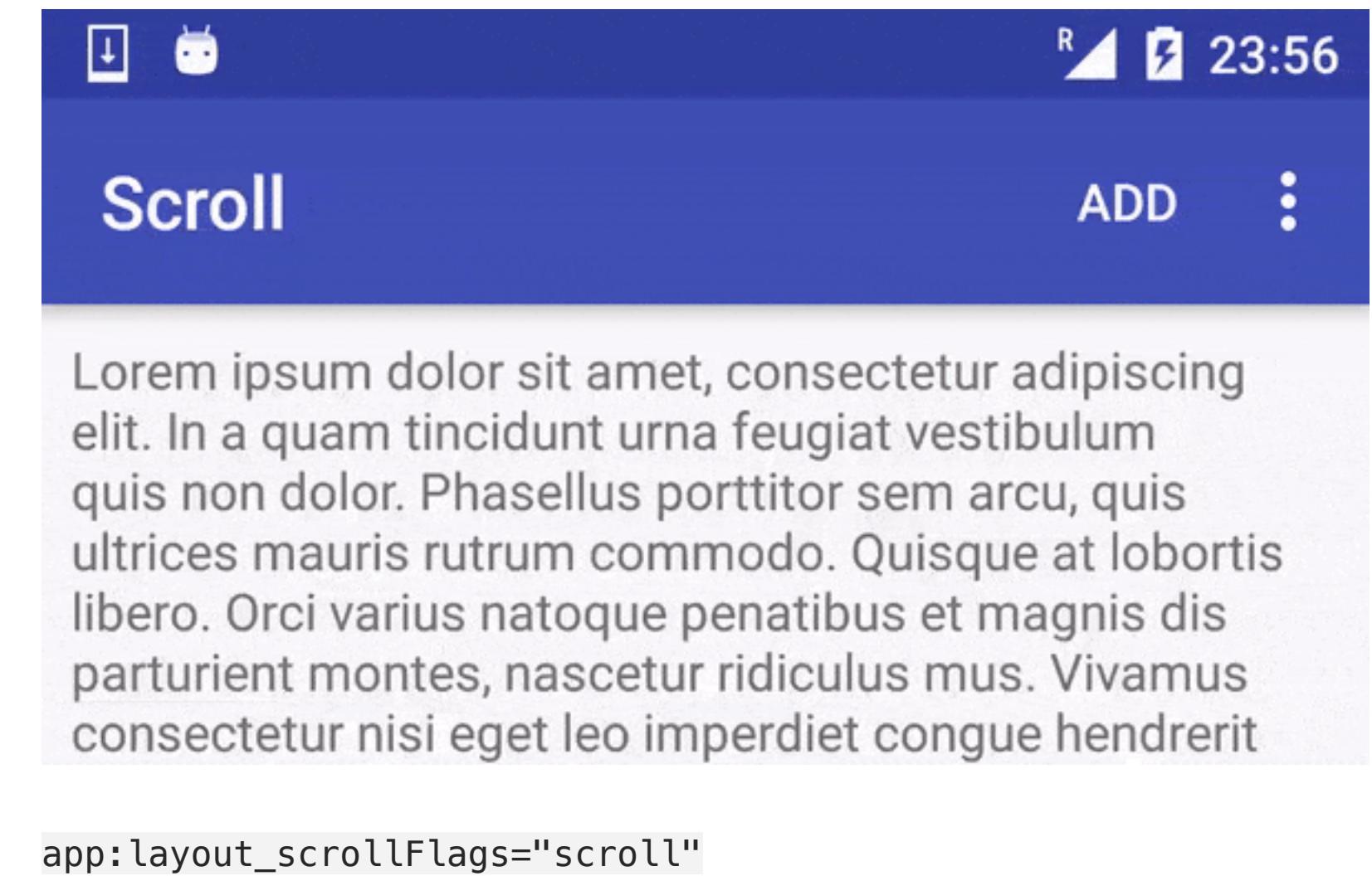
AppBarLayout

AppBarLayout - вертикальный LinearLayout, обеспечивающий поведение своих дочерних элементов во время скроллинга. Поведение дочерних элементов определяется через атрибут app:layout_scrollFlags.

```
<androidx.coordinatorlayout.widget.CoordinatorLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:fitsSystemWindows="true">  
  
    <com.google.android.material.appbar.AppBarLayout  
        android:id="@+id/app_bar"  
        android:layout_width="match_parent"  
        android:layout_height="@dimen/app_bar_height"  
        android:fitsSystemWindows="true"  
        android:theme="@style/AppTheme.AppBarOverlay">  
  
        <com.google.android.material.appbar.CollapsingToolbarLayout  
            android:id="@+id/toolbar_layout"  
            android:layout_width="match_parent"  
            android:layout_height="match_parent"  
            android:fitsSystemWindows="true"  
            app:contentScrim="?attr/colorPrimary"  
            app:layout_scrollFlags="scroll|exitUntilCollapsed"  
            app:toolbarId="@+id/toolbar">  
  
            <androidx.appcompat.widget.Toolbar  
                android:id="@+id/toolbar"  
                android:layout_width="match_parent"  
                android:layout_height="?attr/actionBarSize"  
                app:layout_collapseMode="pin"  
                app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
        </com.google.android.material.appbar.CollapsingToolbarLayout>  
    </com.google.android.material.appbar.AppBarLayout>  
  
    <include layout="@layout/content_scrolling" />  
  
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

AppBarLayout

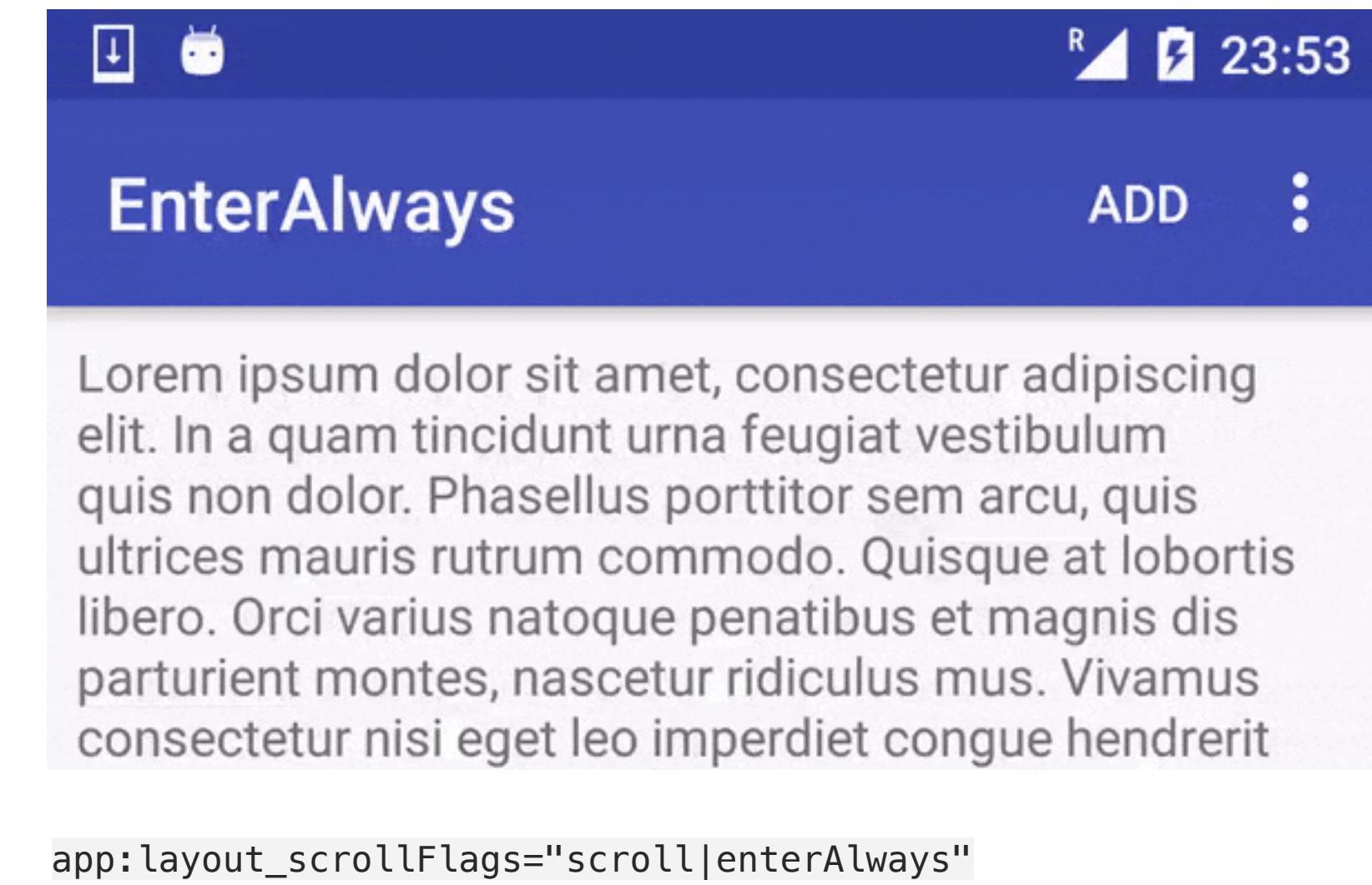
- AppBarLayout - вертикальный LinearLayout, обеспечивающий поведение своих дочерних элементов во время скроллинга. Поведение дочерних элементов определяется через атрибут app:layout_scrollFlags.
 - scroll - дочерний элемент будет вести себя как обычная вью в скопляющемся лэйауте.



AppBarLayout

AppBarLayout - вертикальный LinearLayout, обеспечивающий поведение своих дочерних элементов во время скроллинга. Поведение дочерних элементов определяется через атрибут app:layout_scrollFlags.

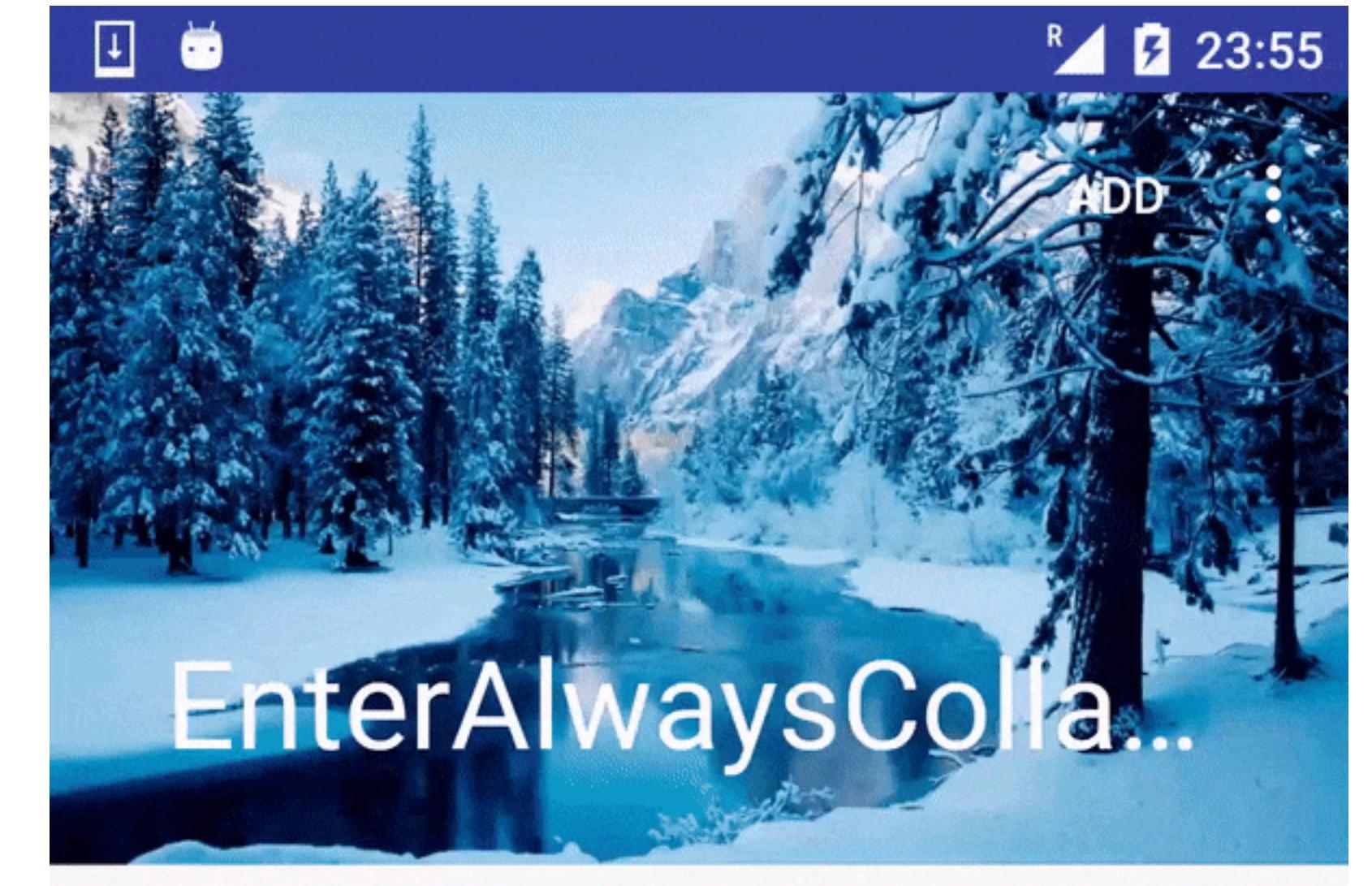
- › scroll - дочерний элемент будет вести себя как обычная вью в скролющемся лэйауте.
- › enterAlways - дочерний элемент скролится вместе со скролющемся контентом, но возвращается на экран



AppBarLayout

AppBarLayout - вертикальный LinearLayout, обеспечивающий поведение своих дочерних элементов во время скроллинга. Поведение дочерних элементов определяется через атрибут app:layout_scrollFlags.

- › scroll - дочерний элемент будет вести себя как обычная вью в скролющемся лэйауте.
- › enterAlways - дочерний элемент скролится вместе со скролющемся контентом, но возвращается на экран
- › enterAlwaysCollapsed - дополнительный флаг, для использования вместе с CollapsingToolbarLayout. При обратном скролле тулбар вернётся в уменьшенном виде, а в конце скролла увеличится

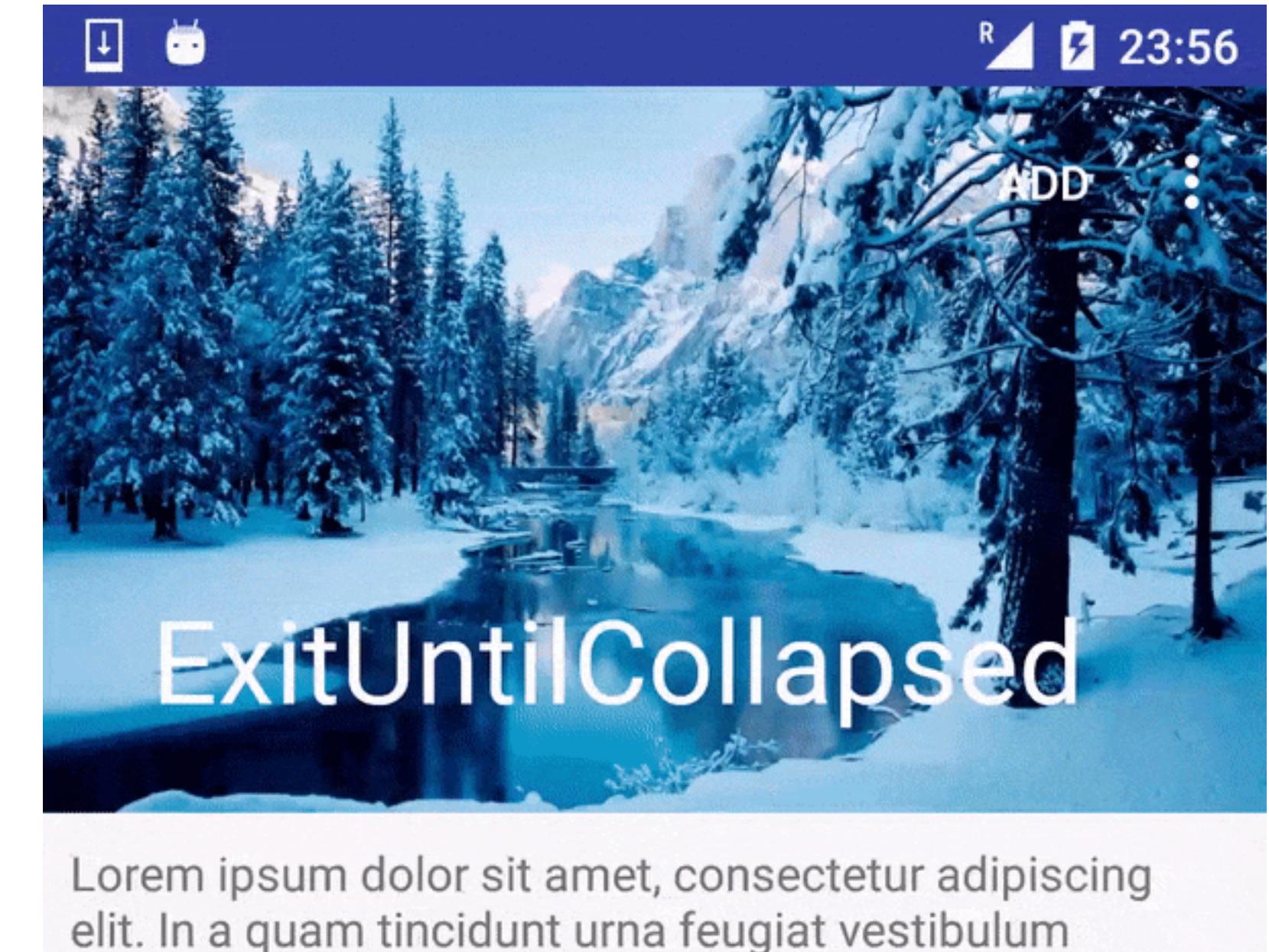


app:layout_scrollFlags="scroll|enterAlways|enterAlwaysCollapsed"

AppBarLayout

AppBarLayout - вертикальный LinearLayout, обеспечивающий поведение своих дочерних элементов во время скроллинга. Поведение дочерних элементов определяется через атрибут app:layout_scrollFlags.

- › exitUntilCollapsed - коллапсит тулбар при скролле, увеличивает когда контент полностью проскоблен назад.

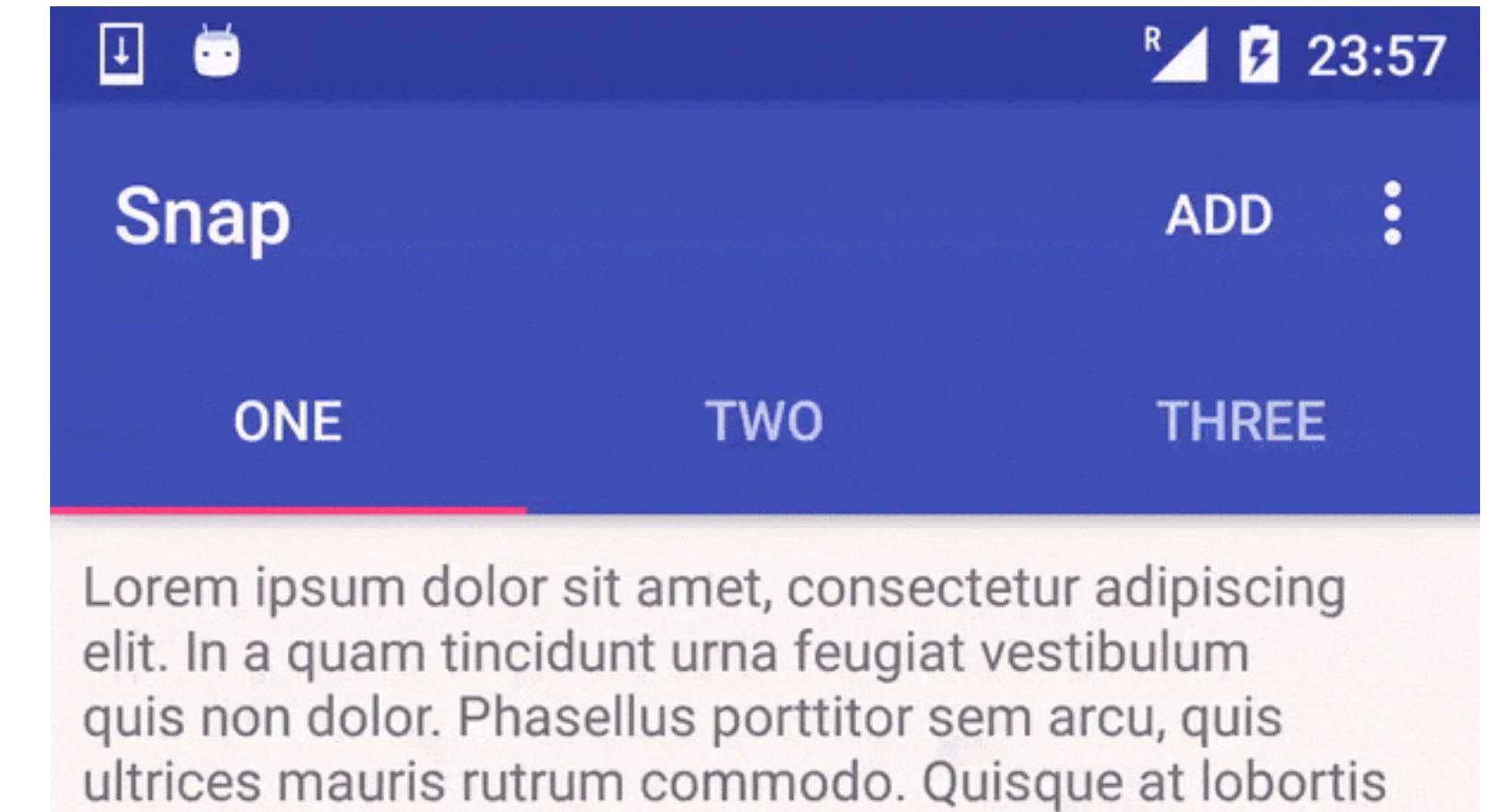


app:layout_scrollFlags="scroll|exitUntilCollapsed"

AppBarLayout

AppBarLayout - вертикальный LinearLayout, обеспечивающий поведение своих дочерних элементов во время скроллинга. Поведение дочерних элементов определяется через атрибут app:layout_scrollFlags.

- exitUntilCollapsed - коллапсит тулбар при скролле, увеличивает когда контент полностью проскоблен назад.
- snap - используется для различных реализаций "прилипчевого" поведения.



```
<android.support.design.widget.AppBarLayout
    android:id="@+id/appBarLayout"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        app:layout_scrollFlags="scroll|snap"
        app:popupTheme="@style/ThemeOverlay.AppCompat.Light" />

    <android.support.design.widget.TabLayout
        android:id="@+id/tabLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        ...
    </android.support.design.widget.TabLayout>
</android.support.design.widget.AppBarLayout>
```

CollapsingToolBarLayout

CollapsingToolBarLayout - обёртка над Toolbar, обеспечивающий визуальный эффект сворачивания панели управления и всех её дочерних элементов во время скроллинга.

```
<androidx.coordinatorlayout.widget.CoordinatorLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:fitsSystemWindows="true">  
  
    <com.google.android.material.appbar.AppBarLayout  
        android:id="@+id/app_bar"  
        android:layout_width="match_parent"  
        android:layout_height="@dimen/app_bar_height"  
        android:fitsSystemWindows="true"  
        android:theme="@style/AppTheme.AppBarOverlay">  
  
        <com.google.android.material.appbar.CollapsingToolbarLayout  
            android:id="@+id/toolbar_layout"  
            android:layout_width="match_parent"  
            android:layout_height="match_parent"  
            android:fitsSystemWindows="true"  
            app:contentScrim="?attr/colorPrimary"  
            app:layout_scrollFlags="scroll|exitUntilCollapsed"  
            app:toolbarId="@+id/toolbar">  
  
            <androidx.appcompat.widget.Toolbar  
                android:id="@+id/toolbar"  
                android:layout_width="match_parent"  
                android:layout_height="?attr/actionBarSize"  
                app:layout_collapseMode="pin"  
                app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
        </com.google.android.material.appbar.CollapsingToolbarLayout>  
    </com.google.android.material.appbar.AppBarLayout>  
  
    <include layout="@layout/content_scrolling" />  
  
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

CollapsingToolBarLayout

CollapsingToolBarLayout - обёртка над Toolbar, обеспечивающий визуальный эффект сворачивания панели управления и всех её дочерних элементов во время скроллинга.

```
<androidx.coordinatorlayout.widget.CoordinatorLayout  
    xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:fitsSystemWindows="true">  
  
    <com.google.android.material.appbar.AppBarLayout  
        android:id="@+id/app_bar"  
        android:layout_width="match_parent"  
        android:layout_height="@dimen/app_bar_height"  
        android:fitsSystemWindows="true"  
        android:theme="@style/AppTheme.AppBarOverlay">  
  
        <com.google.android.material.appbar.CollapsingToolbarLayout  
            android:id="@+id/toolbar_layout"  
            android:layout_width="match_parent"  
            android:layout_height="match_parent"  
            android:fitsSystemWindows="true"  
            app:contentScrim="?attr/colorPrimary"  
            app:layout_scrollFlags="scroll|exitUntilCollapsed"  
            app:toolbarId="@+id/toolbar">  
  
            <androidx.appcompat.widget.Toolbar  
                android:id="@+id/toolbar"  
                android:layout_width="match_parent"  
                android:layout_height="?attr/actionBarSize"  
                app:layout_collapseMode="pin"  
                app:popupTheme="@style/AppTheme.PopupOverlay" />  
  
        </com.google.android.material.appbar.CollapsingToolbarLayout>  
    </com.google.android.material.appbar.AppBarLayout>  
  
    <include layout="@layout/content_scrolling" />  
  
</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

CollapsingToolBarLayout

CollapsingToolBarLayout - обёртка над Toolbar, обеспечивающий визуальный эффект сворачивания панели управления и всех её дочерних элементов во время скроллинга.

- toolbarId - id чайлда, который является тулбаром
- title - тайтл, размер которого будет меняться.
- contentScrim - реализует обрезку холста и его заливку переданным значением атрибута.
- statusBarScrim - реализует заливку statusbar переданным значением атрибута.
- layout_collapseMode - обеспечивает различное поведение дочернего элемента при коллапсинге.

```
<androidx.coordinatorlayout.widget.CoordinatorLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true">

    <com.google.android.material.appbar.AppBarLayout
        android:id="@+id/app_bar"
        android:layout_width="match_parent"
        android:layout_height="@dimen/app_bar_height"
        android:fitsSystemWindows="true"
        android:theme="@style/AppTheme.AppBarOverlay">

        <com.google.android.material.appbar.CollapsingToolbarLayout
            android:id="@+id/toolbar_layout"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:fitsSystemWindows="true"
            app:contentScrim="?attr/colorPrimary"
            app:layout_scrollFlags="scroll|exitUntilCollapsed"
            app:toolbarId="@+id/toolbar">

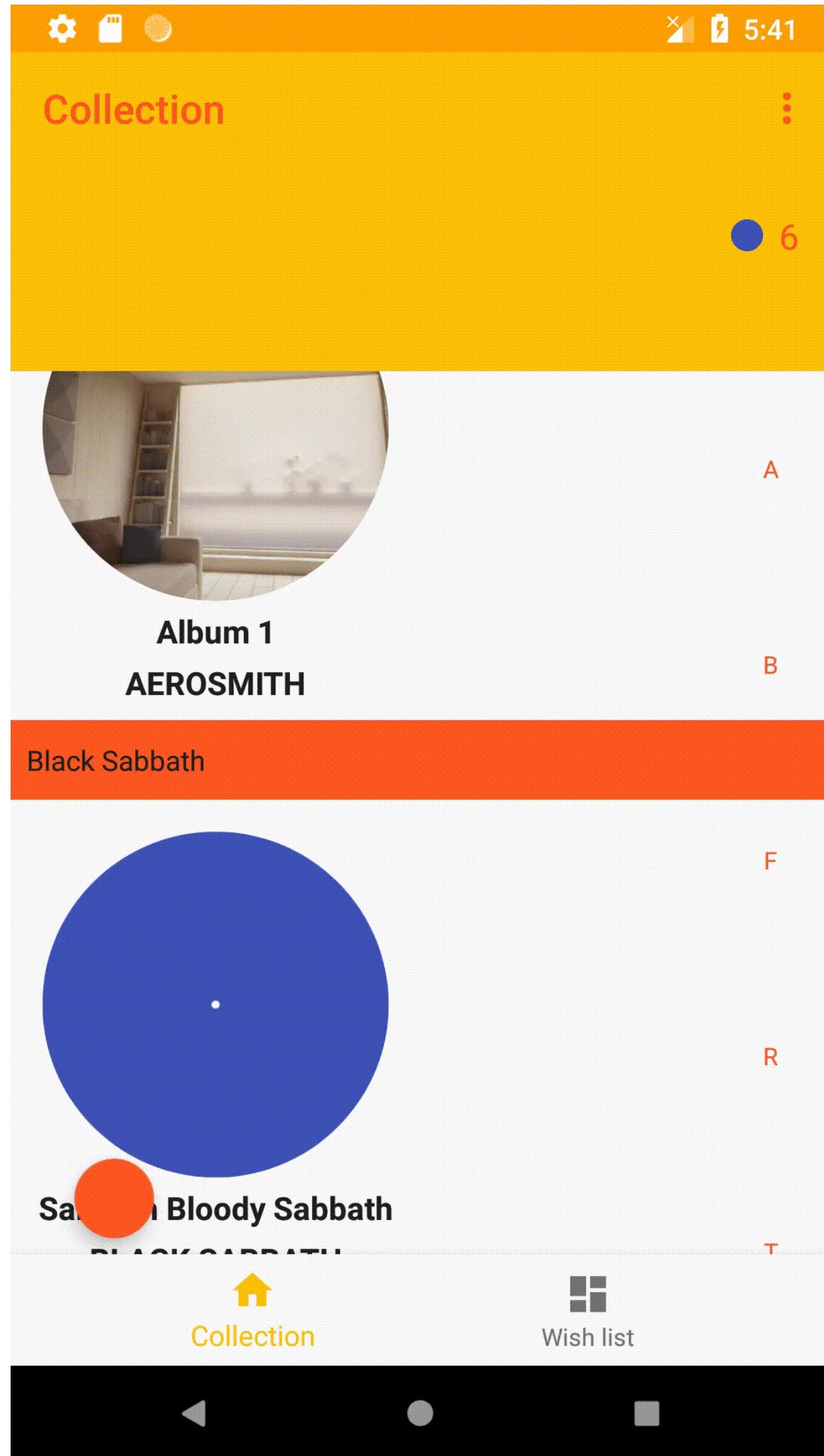
            <androidx.appcompat.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                app:layout_collapseMode="pin"
                app:popupTheme="@style/AppTheme.PopupOverlay" />

            </com.google.android.material.appbar.CollapsingToolbarLayout>
        </com.google.android.material.appbar.AppBarLayout>

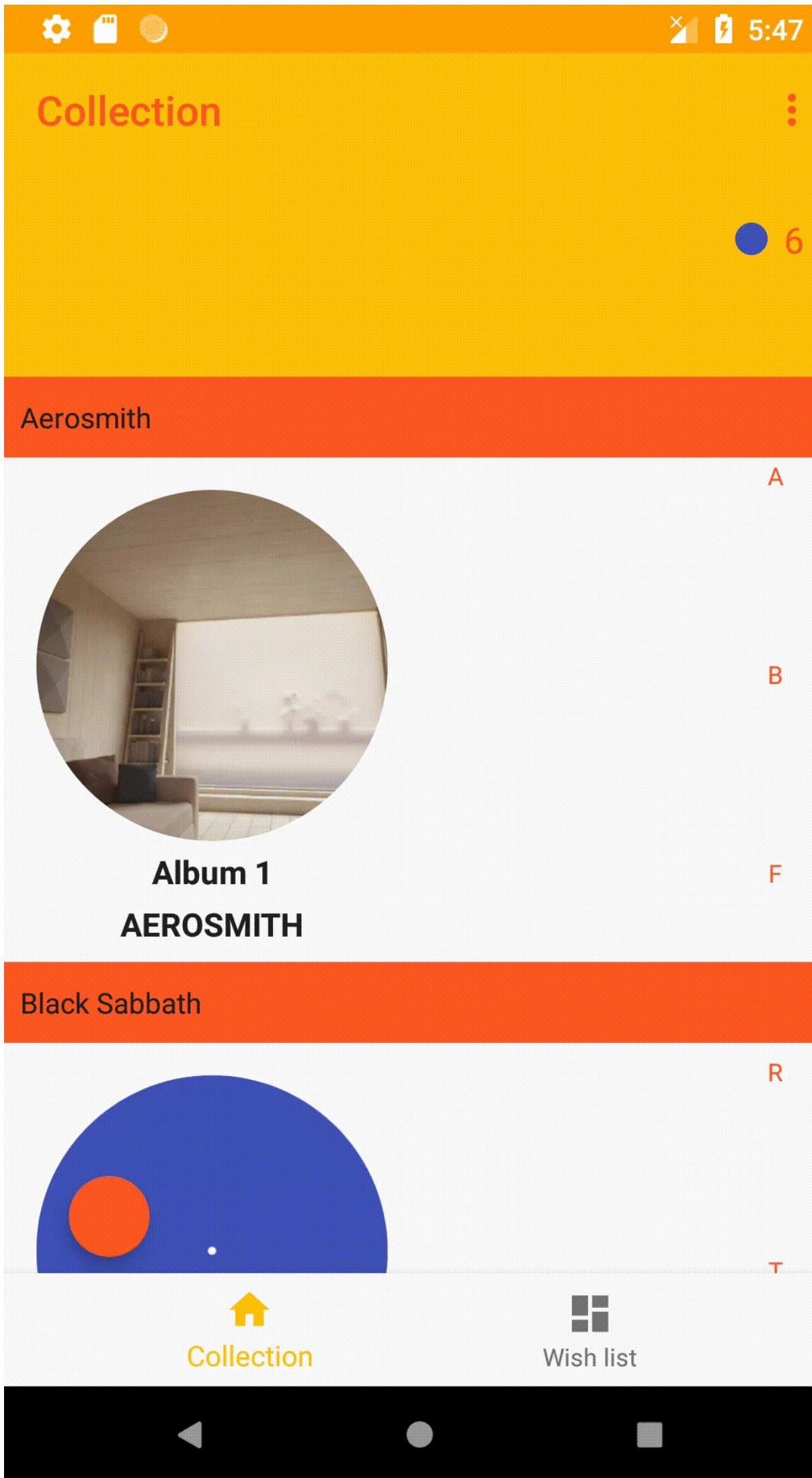
        <include layout="@layout/content_scrolling" />

    </androidx.coordinatorlayout.widget.CoordinatorLayout>
```

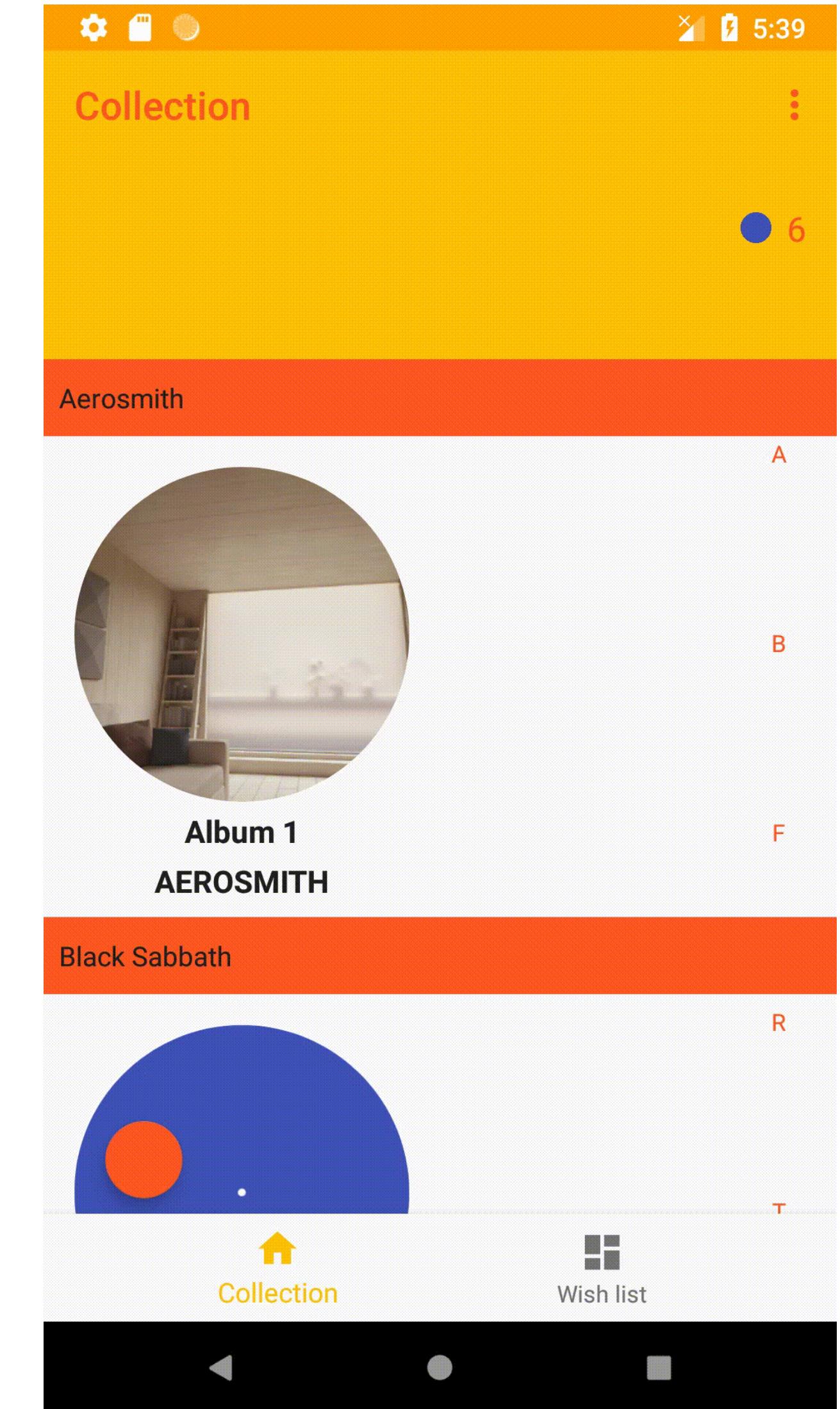
CollapsingToolBarLayout



`layout_collapseMode="none"`



`layout_collapseMode="pin"`



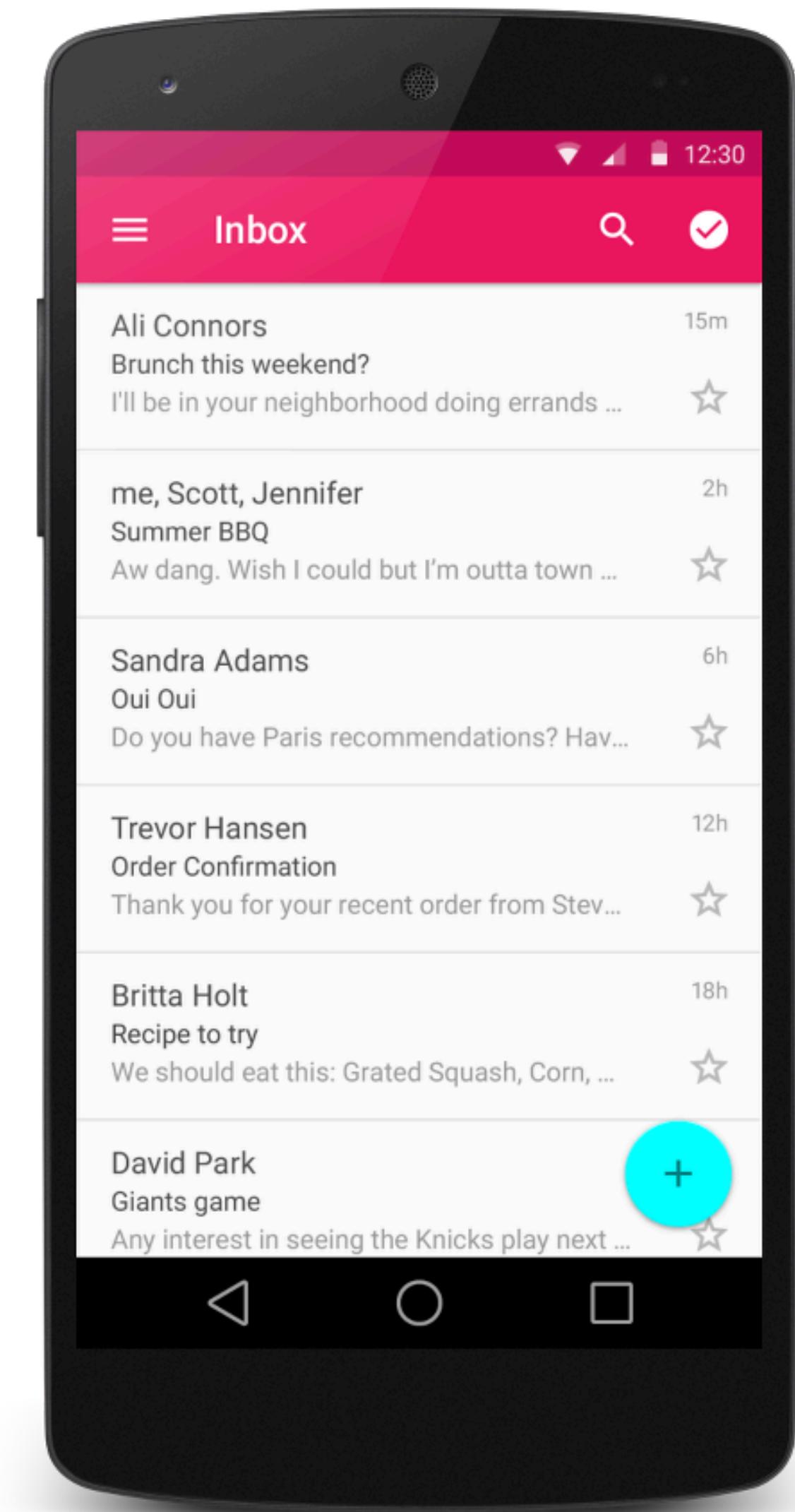
`layout_collapseMode="parallax"`

RecyclerView



RecyclerView

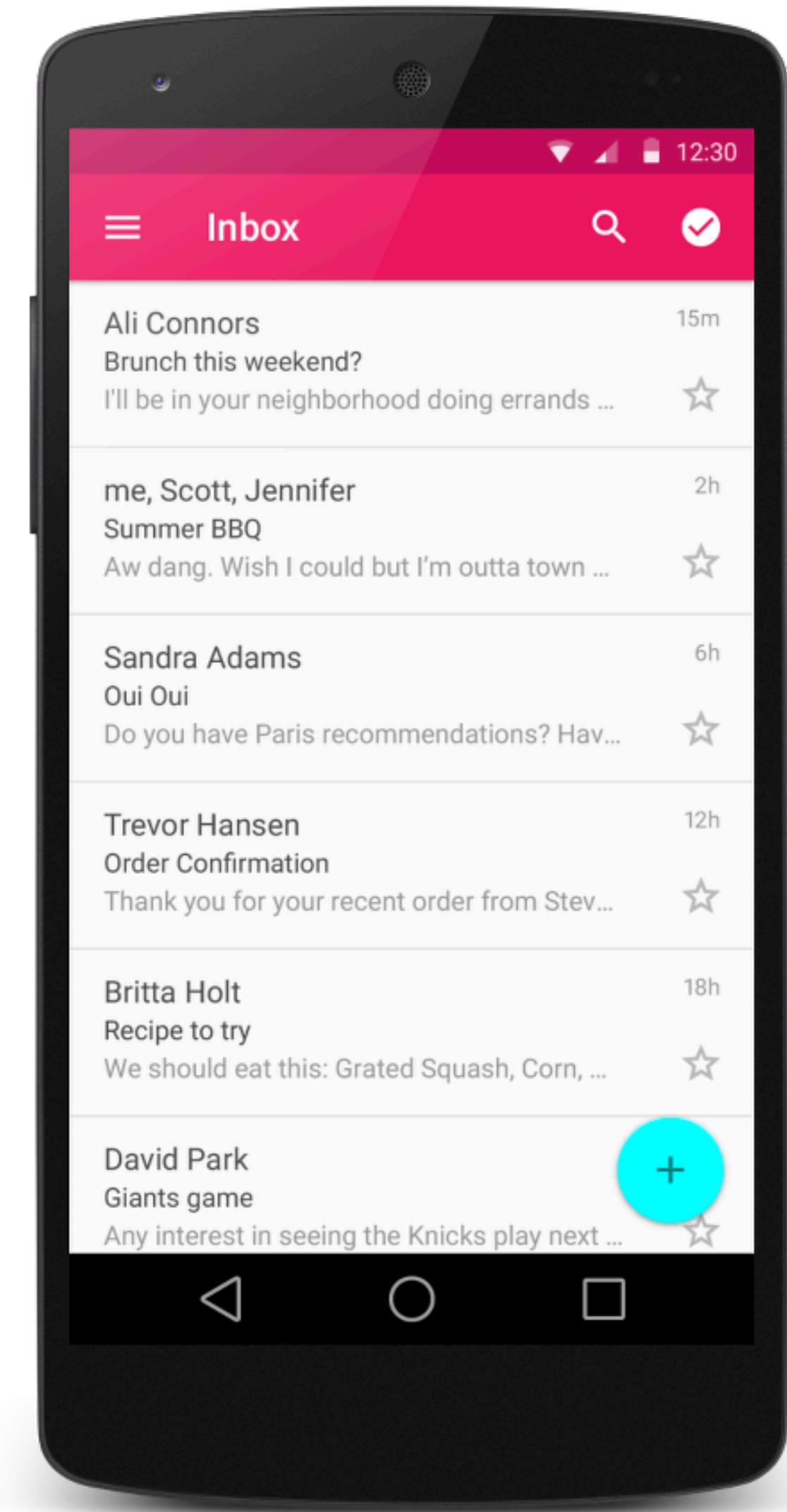
Предназначен для отображения больших наборов данных, которые можно эффективно прокручивать, сохраняя при этом ограниченное количество представлений.



RecyclerView

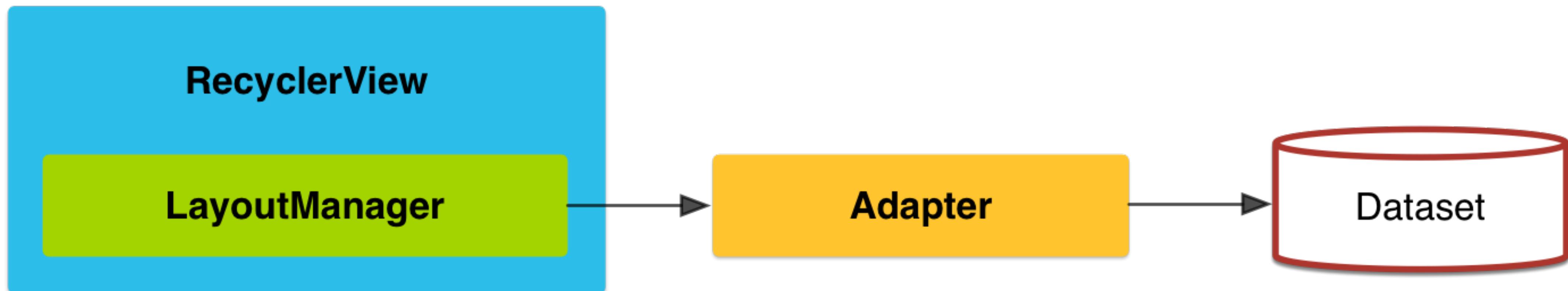
Предназначен для отображения больших наборов данных, которые можно эффективно прокручивать, сохраняя при этом ограниченное количество представлений.

Рекомендуется использовать в случаях, когда имеются коллекции данных, элементы которых изменяются во время выполнения в зависимости от действий пользователя или сетевых событий.



RecyclerView

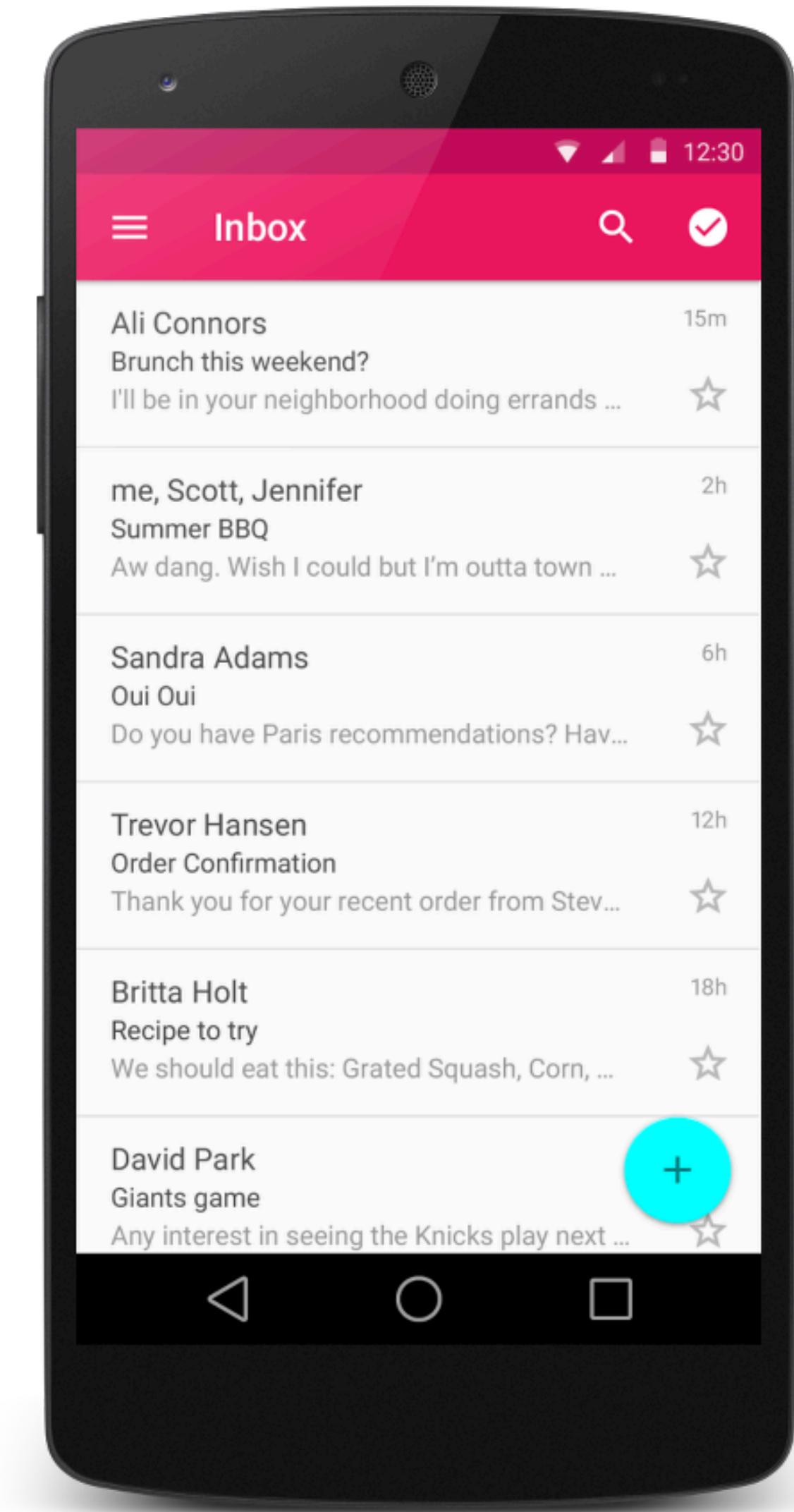
Для работы с RecyclerView, необходимо определить адаптер и менеджер макетов. Для создания адаптера необходимо наследовать класс RecyclerView.Adapter. Характеристики реализации зависят от особенностей используемого набора данных и типа представлений.



RecyclerView

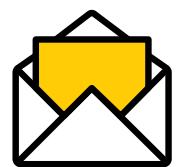
Менеджер макетов размещает представления элемента внутри RecyclerView и определяет порядок повторного использования представлений элемента, которые перестают быть видимыми пользователю.

- › LinearLayoutManager для отображения элементов в виде списка с вертикальной или горизонтальной прокруткой;
- › GridLayoutManager для отображения элементов в виде сетки;
- › StaggeredGridLayoutManager для отображения элементов в виде шахматной сетки.

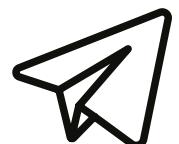


Спасибо за внимание

Степан Чурюканов



tewnn@yandex.ru



@tewnn