

Material Design, ПОЛЬЗОВАТЕЛЬСКИЙ ИНТЕРФЕЙС



Material Design

Material Design – дизайн система/подход, который позволяет создавать консистентный пользовательский опыт на всех экранах: десктоп, смартфон, часы, планшеты, телевизоры и т.д.

Четыре принципа Material Design

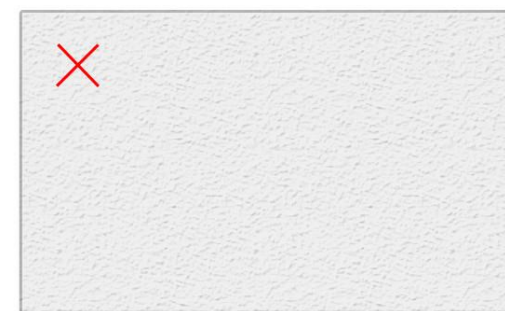
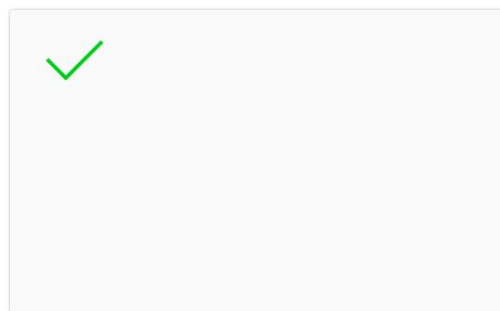
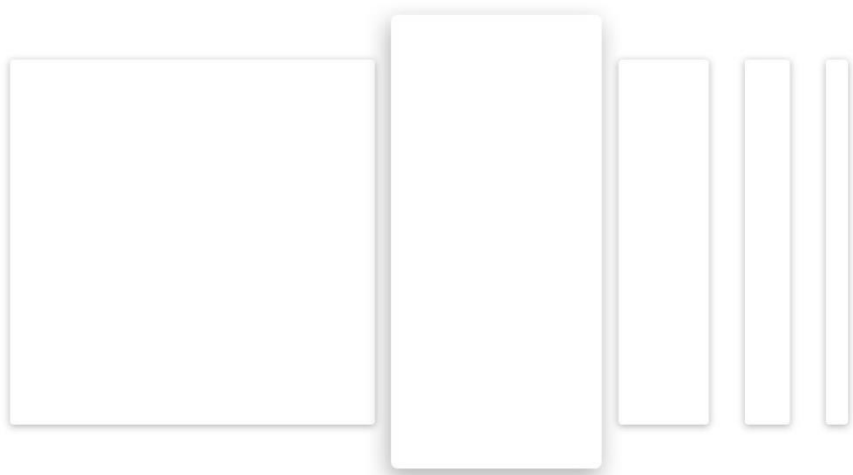
- Тактильные поверхности
- Полиграфический дизайн
- Осмысленная анимация
- Адаптивный дизайн

Тактильные поверхности

Поверхность:

Интерфейс складывается из осязаемых слоев «цифровой бумаги» - тактильных поверхностей. Слои расположены на разной высоте и отбрасывают тени друг на друга.

Т.е. поверхность – «контейнер» с тенью и ничего больше.



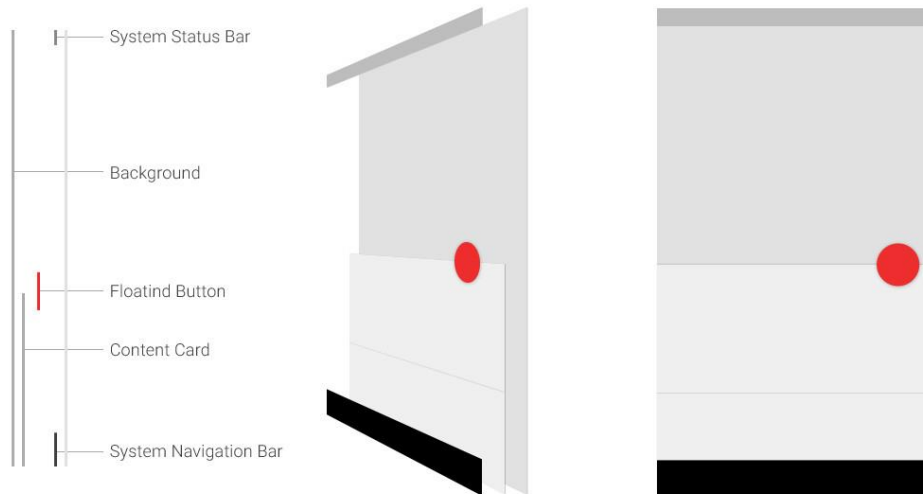
Тактильные поверхности

Глубина:

Глубина помогает сфокусировать внимание пользователя на критически важных вещах. Также глубина задает подсказки о взаимодействии.

Вопрос:

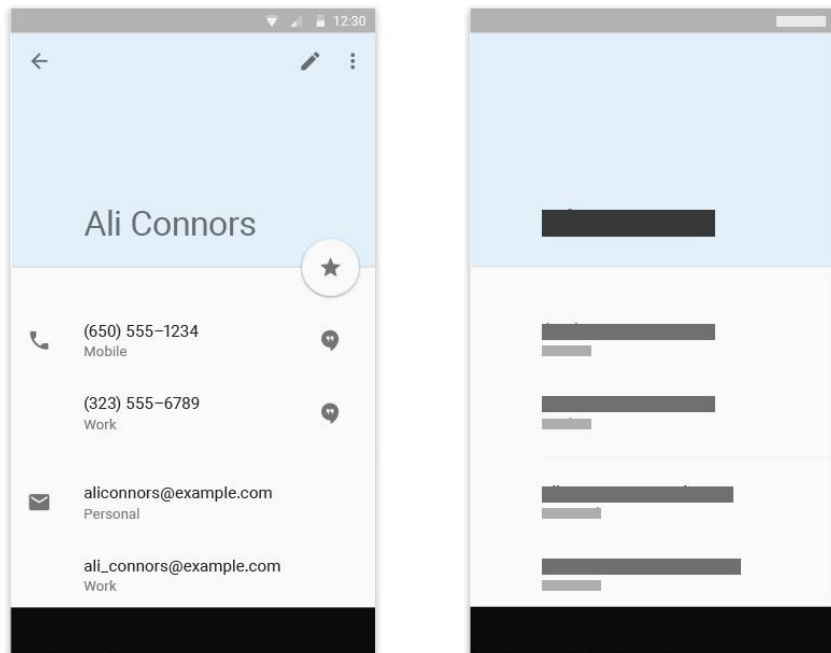
Что делать с темной темой?



Полиграфический дизайн

Изящная типографика:

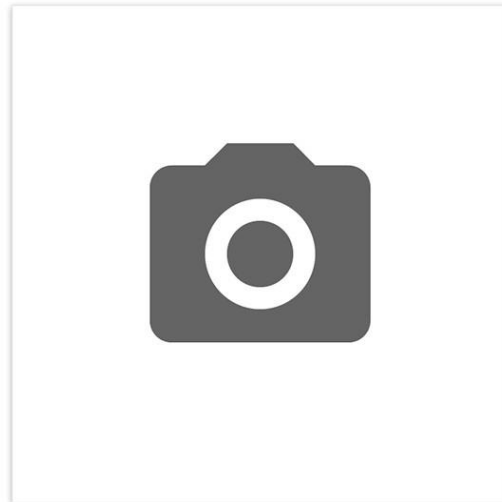
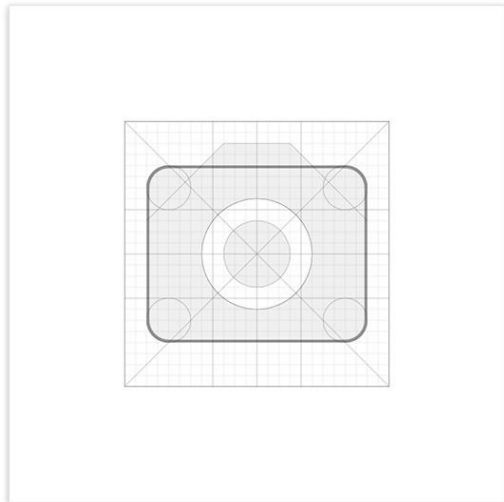
Типографика играет важную роль. Выбор и композиция шрифта задает стиль издания (в терминах газет). Также типографика задает структуру контента.



Полиграфический дизайн

Геометрическая иконографика:

Простые иконки в Android использовались давно, но в Material Design они стали еще проще.



Полиграфический дизайн

Цвет:

В Material Design стандартная цветовая палитра состоит из основного и акцентного цветов. Основной – для больших областей вроде Action Bar. Более яркий акцентный используется точно в элементах управления, призван привлекать внимание к ключевым элементам.





Осмысленные анимации

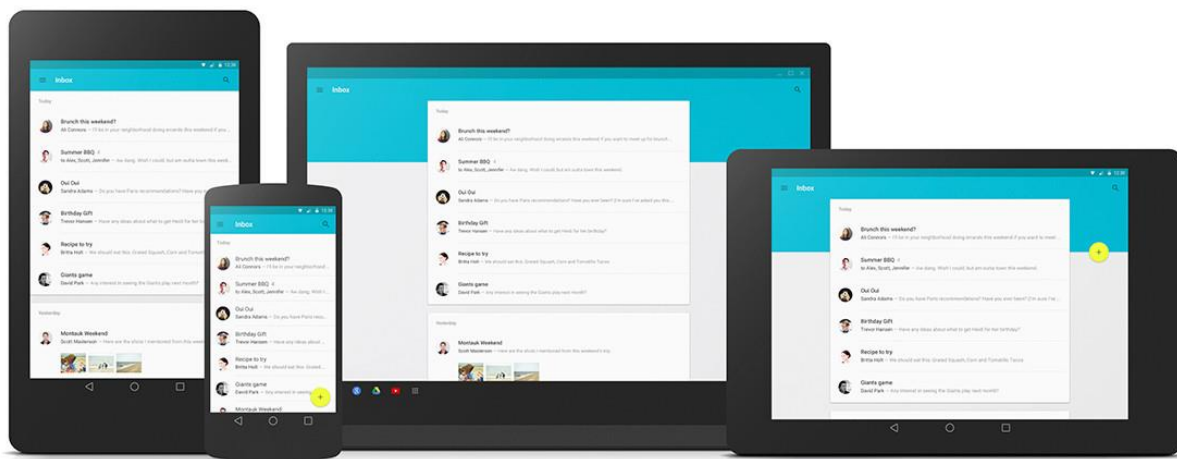
В реальном мире объекты не могут появиться из ниоткуда или исчезнуть в никуда. Поэтому в Material Design анимация используется, чтобы показать, что произошло. Активное движение привлекает взгляд, т.е. анимацией мы управляем вниманием пользователя.

Принципы:

- Асимметрия
- Реакция
- Микроанимации
- Четкость и резкость

Адаптивный дизайн

Адаптивный дизайн – то, как мы можем применить три первые концепции на разных устройствах и экранах разных форм-факторов.



Material Design

- В 2014 году, вместе с выходом Android 5.0 Lollipop, была представлена концепция Material Design.
- В 2018 году вышла вторая версия Material. Добавили больше возможностей кастомизации.
(Обновленная цветовая схема, новые гайдлайны работы со шрифтами, поддержка форм/shapes)
- В 2021 году представили концепцию Material You. Цель новой концепции – персонализировать пользовательский опыт. (Обновление цветовой палитры, Dynamic Color, обновление UI компонентов)

Источники:

<https://habr.com/ru/companies/redmadrobot/articles/252773/>

<https://habr.com/ru/companies/surfstudio/articles/653115/>

<https://m2.material.io/>

<https://m3.material.io/>

Основные элементы

View

- TextView
- ImageView
- Button
- EditText
- *ViewGroup*
- ...

TextView

XML

```
<TextView
    android:id="@+id/title"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textColor="@color/black"
    android:textSize="32sp"
    android:textStyle="italic"
    tools:text="Social network" />
```

ViewGroup

- FrameLayout
- LinearLayout
- CounstraintLayout
- NestedScrollView
- ...

FrameLayout

XML

```
<FrameLayout 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"
    xmlns:tools="http://schemas.android.com/tools">

    <androidx.recyclerview.widget.RecyclerView
        <--...-->
        />

</FrameLayout>
```

Тема в Android - XML

Theme

XML

```
<style name="Base.Theme.Nsu_project_xml"
parent="Theme.Material3.Light.NoActionBar">
    <!-- Customize your light theme here. -->
    <!-- <item name="colorPrimary">@color/my_light_primary</item> -->
    <item name="colorPrimary">@color/yellow</item>
    <item name="colorSecondary">@color/blue</item>
    <item name="colorSurface">@color/grey_300</item>
    <item name="colorOnSurface">@color/black</item>
    <item name="android:statusBarColor">@color/black</item>
    <item name="loginButtonColor">@color/grey</item>
</style>
```

Theme

XML

```
<application
    ...
    android:theme="@style/Theme.Nsu_project_xml"
    ...
</application>
```

Тема в Android - Compose

Theme

Kotlin

```
@Composable
fun Nsu_projectTheme(
    darkTheme: Boolean = isSystemInDarkTheme(),
    // Dynamic color is available on Android 12+
    dynamicColor: Boolean = true,
    content: @Composable () -> Unit
) {
    val colorScheme = when {
        dynamicColor && Build.VERSION.SDK_INT >= Build.VERSION_CODES.S -> {
            val context = LocalContext.current
            if (darkTheme) dynamicDarkColorScheme(context) else
dynamicLightColorScheme(context)
        }

        darkTheme -> DarkColorScheme
        else -> LightColorScheme
    }

    MaterialTheme(
        colorScheme = colorScheme,
        typography = Typography,
        content = content,
    )
}
```

Theme

Kotlin

```
class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContent {
            Nsu_projectTheme {
                Scaffold(modifier = Modifier.fillMaxSize()) { innerPadding ->
                    LazyColumn {
                        items(
                            items = brands,
                        ) {
                            //Any composable with any modifiers
                        }
                    }
                }
            }
        }
    }
}
```

Стили в Android - XML

Theme

XML

```
<style name="ThemeOverlay.App.TextInputLayout" parent="">
    <item name="colorPrimary">@color/blue</item>
    <item
name="editTextStyle">@style/Widget.Material3.TextInputEditText.OutlinedBox</item>
</style>
```

Style

XML

```
<com.google.android.material.textfield.TextInputLayout
    ...
    style="@style/Widget.App.TextInputLayout"
    ...
</com.google.android.material.textfield.TextInputLayout>
```

Формы в Android

Style

XML

```
<style name="Widget.App.LoginButton" parent="Widget.Material3.Button">
    <item name="shapeAppearance">@style/ShapeAppearance.App.LoginButton</item>
    <item name="backgroundTint">?attr/loginButtonColor</item>
</style>

<style name="ShapeAppearance.App.LoginButton"
parent="ShapeAppearance.Material3.SmallComponent">
    <item name="cornerRadius">4dp</item>
</style>
```

Style

Kotlin

```
@Composable
fun Greeting(name: String, modifier: Modifier = Modifier) {
    Text(
        text = "Hello $name!",
        modifier = modifier.clip(RoundedCornerShape(size = 4.dp))
    )
}
```

Принципы работы с темой

Называть цвета своими именами:

Style XML

```
<color name="black">#FF000000</color>
<color name="white">#FFFFFFFF</color>

<color name="yellow">#FFD900</color>
<color name="blue">#0075FF</color>
<color name="grey">#4F4F4F</color>
<color name="grey_300">#F9F9F9</color>
```

Color Kotlin

```
val Purple80 = Color(0xFFD0BCFF)
val PurpleGrey80 = Color(0xFFCCC2DC)
val Pink80 = Color(0xFFE8BFD0)

val Purple40 = Color(0xFF66BB6A)
val PurpleGrey40 = Color(0xFFB2DFDB)
val Pink40 = Color(0xFF81C784)
```

Разносить стили/цвета по файлам:

```
values
  </> attrs.xml
  </> colors.xml
  </> strings.xml
  </> styles.xml
  </> themes.xml
```

```
ui.theme
  Color.kt
  Theme.kt
  Type.kt
```


Принципы работы с темой

Давать осмысленные имена:

Theme

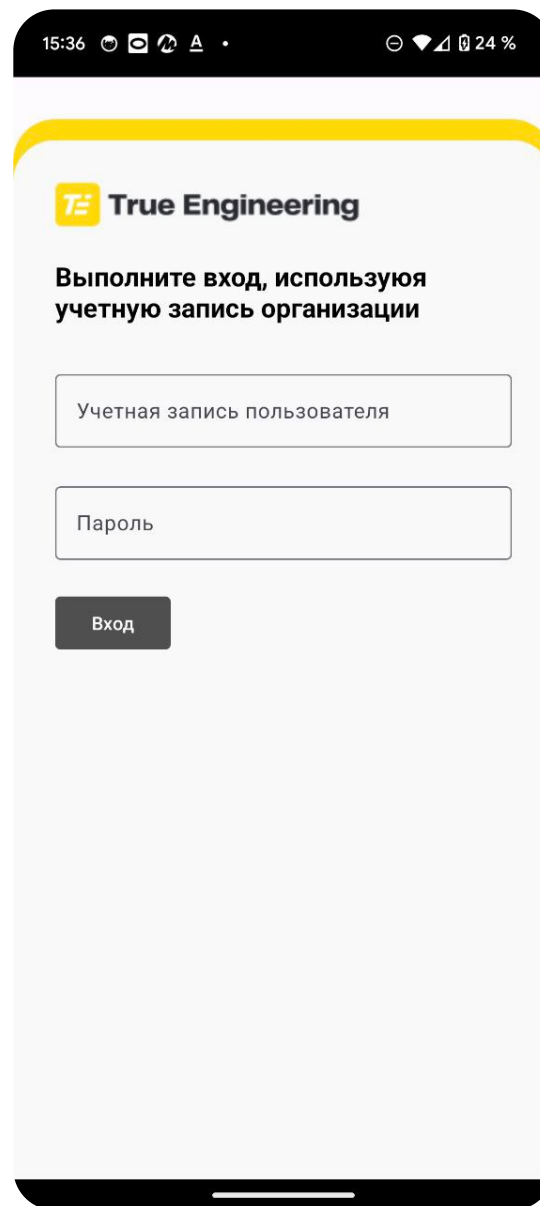
XML

```
<style name="ThemeOverlay.App.TextInputLayout" parent="">
    <item name="colorPrimary">@color/blue</item>
    <item
name="editTextStyle">@style/Widget.Material3.TextInputEditText.OutlinedBox</item>
</style>
```

Практическая работа

Повторить верстку:

- Создание темы
- Переопределение стилей View
- Верстка с использованием разных ViewGroup и View
- Обращение ко View из кода



Домашнее задание

Повторить верстку:

Точное совпадение цветов и размеров не обязательно. Наличие Action Bar сверху не обязательно

Задание со звездочкой:

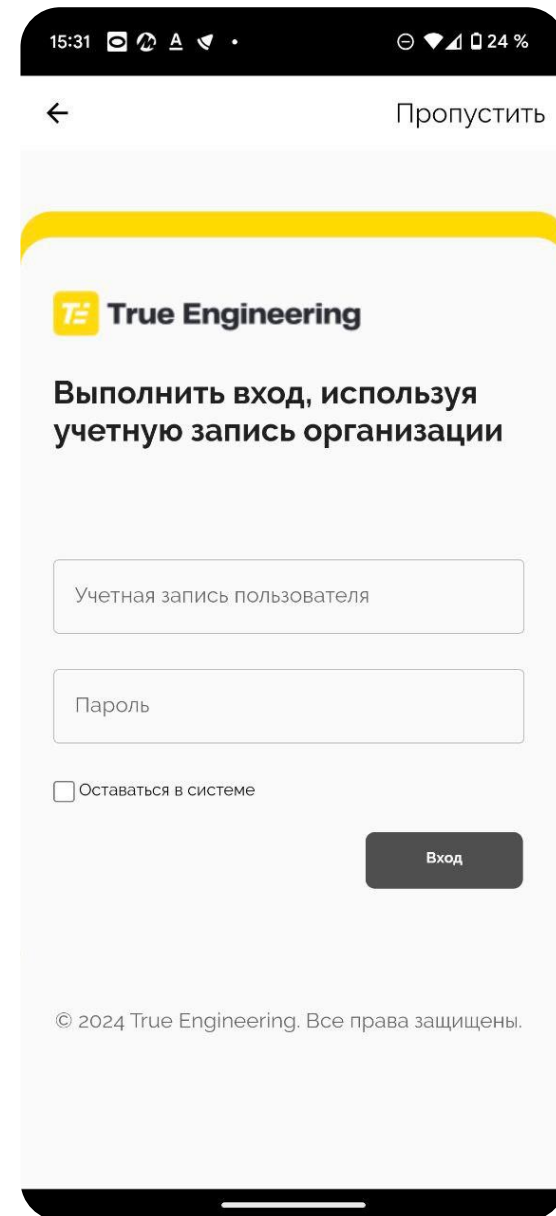
Выполнить первое задание с использованием ConstraintLayout. Т.е. вложенность ViewGroup не должны быть больше одного.

Видео-обзор с объяснением работы Constraint Layout:

<https://www.youtube.com/watch?v=yZYOX6hRYFc>

Документация:

<https://developer.android.com/reference/androidx/constraintlayout/widget/ConstraintLayout>





True Engineering

630128, г. Новосибирск,
ул. Кутателадзе, 4г

(383) 363-33-51, 363-33-50
info@trueengineering.ru
trueengineering.ru

Новосибирский
Государственный
Университет