Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Отчёт по лабораторной работе № 1

Дисциплина: Проектирование мобильных приложений

Тема: Layouts

Выполнил студент гр. 3530901/90202 _____ Т.С. Плетнев
(подпись)

Руководитель _____А.Н. Кузнецов
(подпись)

“____” _____ 2021 г.

Санкт-Петербург
2021

1. Цели работы

- Ознакомиться со средой разработки Android Studio
- Изучить основные принципы верстки layout с использованием View и ViewGroup
- Изучить основные возможности и свойства LinearLayout
- Изучить основные возможности и свойства ConstraintLayout

2. Задачи работы

- Ознакомиться со средой разработки Android Studio
- Изучить основные принципы верстки layout с использованием View и ViewGroup
- Изучить основные возможности и свойства LinearLayout
- Изучить основные возможности и свойства ConstraintLayout

3. Ход работы

3.1 Начало работы

Была поставлена себе задача сверстать первые основные экраны приложения будильник, предварительно создав макеты в Figma (рис. 3.1.1-3.1.2).

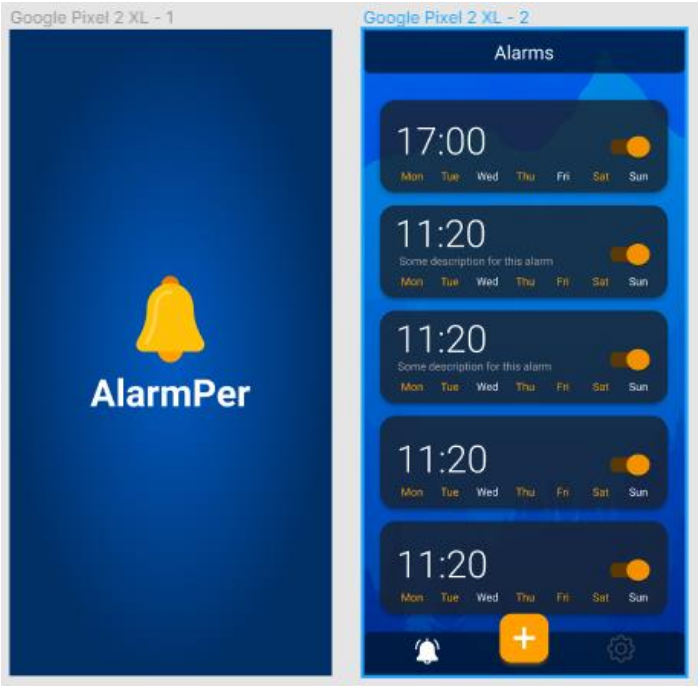

Рис. 3.1.1 SplashScreen и AlarmsList

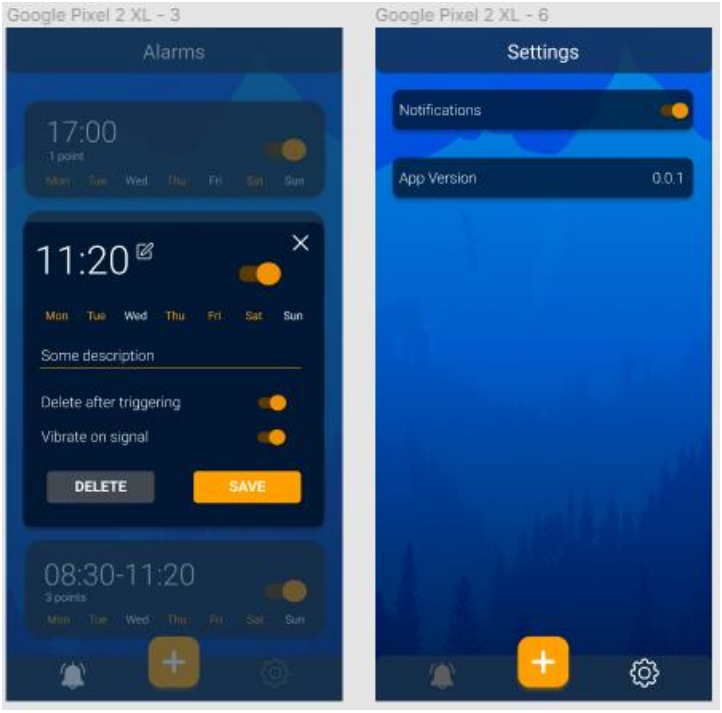

Рис. 3.1.2 AlarmDetails и Settings

Таким образом, были созданы следующие XML-файлы (рис. 3.1.3)

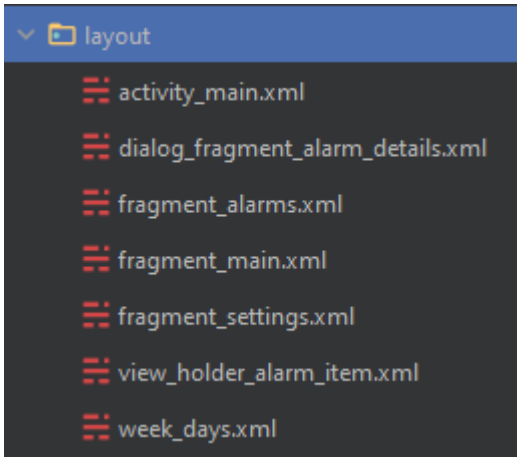

Рис. 3.1.3 Созданные XML файлы

Больше всего нас интересуют файлы dialog_fragment_alarm_details.xml, где был использован ConstraintLayout, а также week_days.xml, который в себе содержит LinearLayout.

3.2 LinearLayout

LinearLayout – ViewGroup, располагающий дочерние элементы в «линию» в нужном направлении. Дополнительный атрибут, раскрывающий возможности этого контейнера – это android:layout_weight. Это атрибут, отвечающий за «вес» элемента в родителе. Для его использования нужно у гибкого параметра размера (ширина/высота) выставить значение 0dp, тем самым дав понять контейнеру, что данный параметр будет зависеть от остальных элементов и будет высчитан позже, исходя из значения веса.

В нашем случае нужно было сверстать часто встречающийся элемент на нескольких экранах – список кликабельных дней недели (CheckedTextView). Поскольку элементы должны быть расположены горизонтально, то у корневого тега задаем значение android:orientation="horizontal", а чтобы layout занимал всю возможную ширину, данную родителем, то выставляем android:layout_width="match_parent". Теперь, чтобы все элементы были расположены равномерно по всей ширине, у каждого выставим android:layout_width="0dp" и android:layout_weight="1". Таким образом, все элементы будут иметь одинаковый вес и контейнер равномерно разделит между ними ширину.

Листинг 3.2.1 week_days.xml

(https://github.com/timatifey/alarmper/blob/dev/alarmper/app/src/main/res/layout/week_days.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <CheckedTextView
        android:id="@+id/monday"
        style="@style/WeekDay"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/Monday_short"
        tools:checked="true" />

    <CheckedTextView
```

```
        android:id="@+id/tuesday"
        style="@style/WeekDay"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/Tuesday_short"
        tools:checked="false" />

    <CheckedTextView
        android:id="@+id/wednesday"
        style="@style/WeekDay"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/Wednesday_short"
        tools:checked="true" />

    <CheckedTextView
        android:id="@+id/thursday"
        style="@style/WeekDay"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/Thursday_short"
        tools:checked="false" />

    <CheckedTextView
        android:id="@+id/friday"
        style="@style/WeekDay"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/Friday_short"
        tools:checked="true" />

    <CheckedTextView
        android:id="@+id/saturday"
        style="@style/WeekDay"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/Saturday_short"
        tools:checked="true" />

    <CheckedTextView
        android:id="@+id/sunday"
        style="@style/WeekDay"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/Sunday_short"
        tools:checked="false" />
</LinearLayout>
```

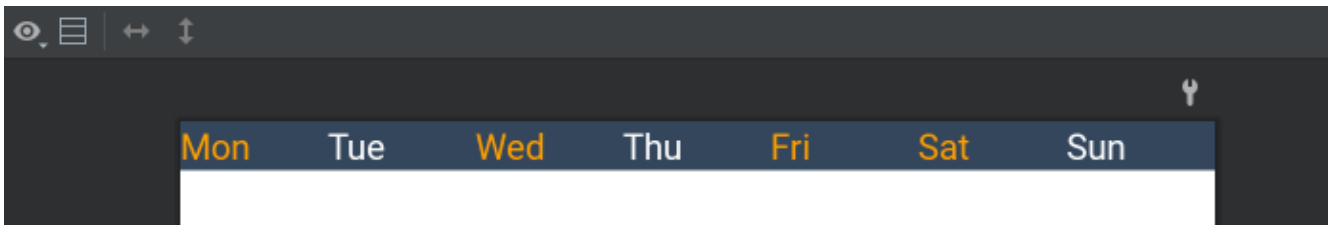

Рис. 3.2.1 Визуальный результат week_days.xml

3.3 ConstraintLayout

ConstraintLayout – контейнер, созданный Google, которого не является частью Android SDK, однако используется повсеместно из-за своих возможностей. Основная проблема, которую решал Google, создав этот инструмент – избавиться от огромной вложенности ViewGroup, поскольку парсинг таких XML файлов в больших проектах занимал долгое время. Основной принцип отношений между дочерними элементами – связь или цепь.

Для примера приведем файл верстки диалогового окна с детальной информацией о будильнике: dialog_fragment_alarm_details.xml (рис. 3.3.1)

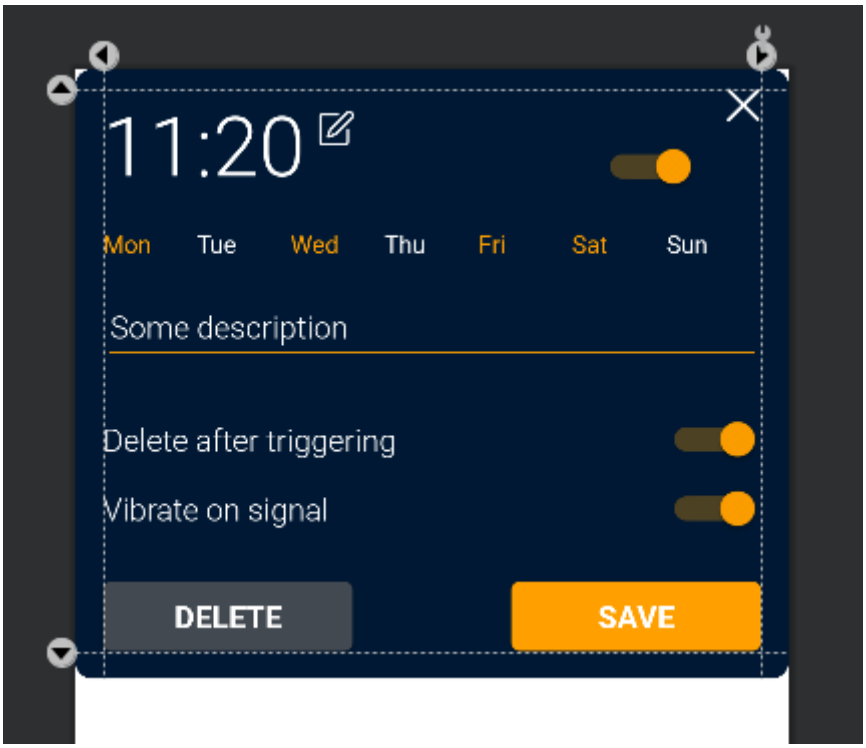

Рис. 3.3.1 Визуальный результат dialog_fragment_alarm_details.xml

Листинг 3.3.1 dialog_fragment_alarm_details.xml

(https://github.com/timatifey/alarmper/blob/dev/alarmper/app/src/main/res/layout/dialog_fragment_alarm_details.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
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:layout_width="match_parent"
    android:layout_height="wrap_content"
```

```
    android:background="@drawable/background_alarm_details">

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline_start"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_begin="16dp" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline_end"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        app:layout_constraintGuide_end="16dp" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline_top"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_begin="11dp" />

    <androidx.constraintlayout.widget.Guideline
        android:id="@+id/guideline_bottom"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        app:layout_constraintGuide_end="16dp" />

    <TextView
        android:id="@+id/alarm_time"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:fontFamily="@font/roboto_light"
        android:textColor="@color/white"
        android:textSize="48sp"
        app:layout_constraintStart_toStartOf="@id/guideline_start"
        app:layout_constraintTop_toTopOf="@id/guideline_top"
        tools:text="11:20" />

    <ImageView
        android:id="@+id/ic_edit"
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:layout_marginStart="8dp"
        android:src="@drawable/ic_edit"
        app:layout_constraintBottom_toBottomOf="@id/alarm_time"
        app:layout_constraintStart_toEndOf="@id/alarm_time"
        app:layout_constraintTop_toTopOf="@id/alarm_time"
        app:layout_constraintVertical_bias="0.3" />

    <ImageView
        android:id="@+id/ic_close"
        android:layout_width="20dp"
        android:layout_height="20dp"
        android:layout_marginStart="8dp"
        android:src="@drawable/ic_close"
```

```
        app:layout_constraintEnd_toEndOf="@id/guideline_end"
        app:layout_constraintTop_toTopOf="@id/guideline_top" />

    <androidx.appcompat.widget.SwitchCompat
        android:id="@+id/alarm_switcher"
        android:layout_width="50dp"
        android:layout_height="30dp"
        app:layout_constraintBaseline_toBaselineOf="@id/alarm_time"
        app:layout_constraintEnd_toEndOf="@id/guideline_end"
        app:layout_constraintHorizontal_bias="0.8"
        app:layout_constraintStart_toEndOf="@id/ic_edit"
        app:layout_constraintTop_toBottomOf="@id/ic_close"
        tools:checked="true" />

    <include
        android:id="@+id/week_days"
        layout="@layout/week_days"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        app:layout_constraintBottom_toTopOf="@id/alarm_description"
        app:layout_constraintEnd_toEndOf="@id/guideline_end"
        app:layout_constraintStart_toStartOf="@id/guideline_start"
        app:layout_constraintTop_toBottomOf="@id/alarm_time" />

    <EditText
        android:id="@+id/alarm_description"
        style="@style/Text"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:backgroundTint="@color/orange"
        android:fontFamily="@font/roboto_light"
        android:hint="@string/description"
        android:inputType="textMultiLine"
        android:maxLength="150"
        android:maxLines="3"
        android:paddingStart="4dp"
        android:paddingEnd="4dp"
        android:textColor="@color/white"
        android:textColorHighlight="@color/orange"
        android:textColorHint="@color/gray"
        android:textColorLink="@color/orange"
        android:textSize="18sp"
        app:layout_constraintEnd_toEndOf="@id/guideline_end"
        app:layout_constraintStart_toStartOf="@id/guideline_start"
        app:layout_constraintTop_toBottomOf="@id/week_days"
        tools:text="Some description" />

    <TextView
        android:id="@+id/delete_after"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="30dp"
        android:fontFamily="@font/roboto_light"
        android:text="@string/delete_after_triggering"
        android:textColor="@color/white"
        android:textSize="18sp"
```

```
        app:layout_constraintEnd_toStartOf="@id/delete_after_switcher"
        app:layout_constraintStart_toStartOf="@id/guideline_start"
        app:layout_constraintTop_toBottomOf="@id/alarm_description" />

    <androidx.appcompat.widget.SwitchCompat
        android:id="@+id/delete_after_switcher"
        android:layout_width="50dp"
        android:layout_height="30dp"
        app:layout_constraintBottom_toBottomOf="@id/delete_after"
        app:layout_constraintEnd_toEndOf="@id/guideline_end"
        app:layout_constraintHorizontal_bias="1"
        app:layout_constraintStart_toEndOf="@id/delete_after"
        app:layout_constraintTop_toTopOf="@id/delete_after"
        tools:checked="true" />

    <TextView
        android:id="@+id/vibrate"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="16dp"
        android:fontFamily="@font/roboto_light"
        android:text="@string/vibrate_on_signal"
        android:textColor="@color/white"
        android:textSize="18sp"
        app:layout_constraintEnd_toStartOf="@id/vibrate_switcher"
        app:layout_constraintStart_toStartOf="@id/guideline_start"
        app:layout_constraintTop_toBottomOf="@id/delete_after" />

    <androidx.appcompat.widget.SwitchCompat
        android:id="@+id/vibrate_switcher"
        android:layout_width="50dp"
        android:layout_height="30dp"
        app:layout_constraintBottom_toBottomOf="@id/vibrate"
        app:layout_constraintEnd_toEndOf="@id/guideline_end"
        app:layout_constraintHorizontal_bias="1"
        app:layout_constraintStart_toEndOf="@id/vibrate"
        app:layout_constraintTop_toTopOf="@id/vibrate"
        tools:checked="true" />

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/delete_btn"
        style="@style/WhiteColorTextButton"
        android:layout_width="0dp"
        android:layout_height="40dp"
        android:layout_marginTop="30dp"
        android:background="@drawable/background_gray_btn"
        android:text="@string/delete"
        app:layout_constraintBottom_toBottomOf="@id/guideline_bottom"
        app:layout_constraintEnd_toStartOf="@id/save_btn"
        app:layout_constraintHorizontal_chainStyle="spread_inside"
        app:layout_constraintStart_toStartOf="@id/guideline_start"
        app:layout_constraintTop_toBottomOf="@id/vibrate"
        app:layout_constraintWidth_percent="0.35" />

    <androidx.appcompat.widget.AppCompatButton
        android:id="@+id/save_btn"
        style="@style/WhiteColorTextButton"
        android:layout_width="0dp"
```

```
        android:layout_height="40dp"
        android:layout_marginTop="30dp"
        android:background="@drawable/background_orange_btn"
        android:text="@string/save"
        app:layout_constraintBottom_toBottomOf="@id/guideline_bottom"
        app:layout_constraintEnd_toEndOf="@id/guideline_end"
        app:layout_constraintStart_toEndOf="@id/delete_btn"
        app:layout_constraintTop_toBottomOf="@id/vibrate"
        app:layout_constraintWidth_percent="0.35" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

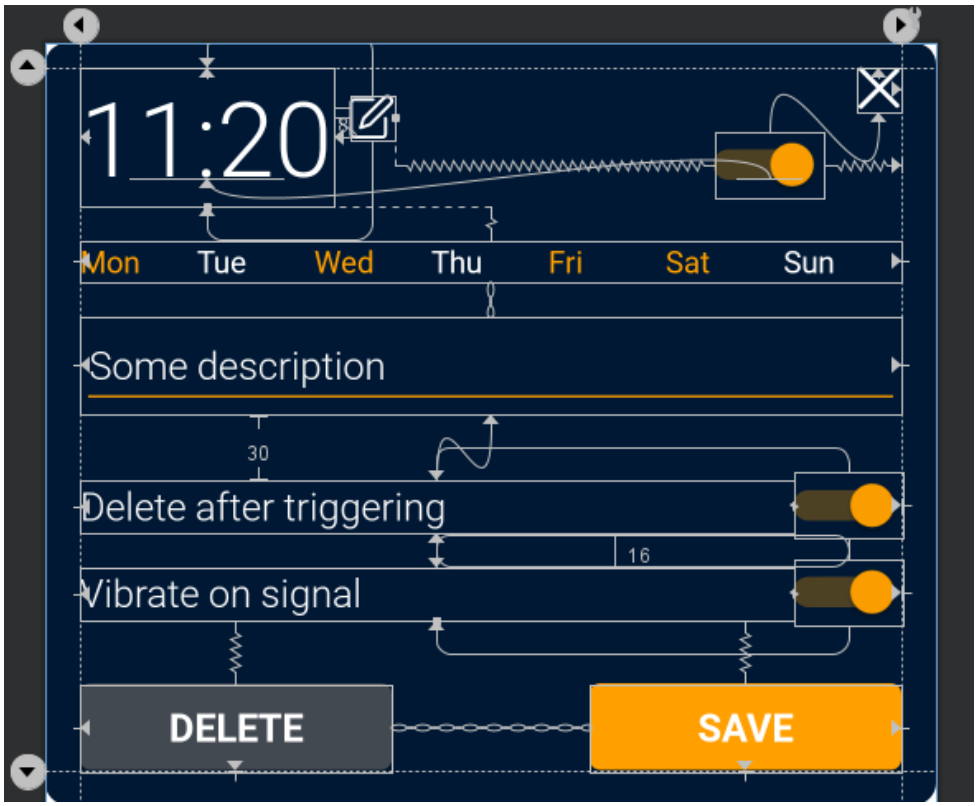

Рис. 3.3.2 Связи в dialog_fragment_alarm_details.xml

Для примера возьмем Switch с id = alarm_switcher:

```
<androidx.appcompat.widget.SwitchCompat
    android:id="@+id/alarm_switcher"
    android:layout_width="50dp"
    android:layout_height="30dp"
    app:layout_constraintBaseline_toBaselineOf="@id/alarm_time"
    app:layout_constraintEnd_toEndOf="@id/guideline_end"
    app:layout_constraintHorizontal_bias="0.8"
    app:layout_constraintStart_toEndOf="@id/ic_edit"
    app:layout_constraintTop_toBottomOf="@id/ic_close"
    tools:checked="true" />
```

Рассмотрим всего его связи подробно:

app:layout_constraintBaseline_toBaselineOf="@id/alarm_time" – благодаря данной связи элемент находится на базовой линии текста alarm_time;

app:layout_constraintEnd_toEndOf="@id/guideline_end" – конец элемента привязывается к конечному guideline;

app:layout_constraintStart_toEndOf="@id/ic_edit" – начало элемента привязывается к концу иконки ic_edit;

app:layout_constraintHorizontal_bias="0.8" – поскольку мы привязаны и в начале и к концу чего-либо, то связи превратились в «пружины», жесткость которых можно изменить, поставив значение 0.8, элемент ближе к guideline_end;

app:layout_constraintTop_toBottomOf="@id/ic_close" – сверху привязывем элемент к низу иконки ic_close.

Также рассмотрим edit text для описания будильника:

```
<EditText
    android:id="@+id/alarm_description"
    style="@style/Text"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:backgroundTint="@color/orange"
    android:fontFamily="@font/roboto_light"
    android:hint="@string/description"
    android:inputType="textMultiLine"
    android:maxLength="150"
    android:maxLines="3"
    android:paddingStart="4dp"
    android:paddingEnd="4dp"
    android:textColor="@color/white"
    android:textColorHighlight="@color/orange"
    android:textColorHint="@color/gray"
    android:textColorLink="@color/orange"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="@id/guideline_end"
    app:layout_constraintStart_toStartOf="@id/guideline_start"
    app:layout_constraintTop_toBottomOf="@id/week_days"
    tools:text="Some description" />
```

Здесь стоит обратить внимание на

```
android:layout_width="0dp"
```

Задав значение ширины равным 0dp, мы говорим контейнеру занять все возможное пространство от начального элемента (guideline_start) к конечному элементу (guideline_end) (рис. 3.3.3), привязавшись к ним с помощью

```
app:layout_constraintEnd_toEndOf="@id/guideline_end"
app:layout_constraintStart_toStartOf="@id/guideline_start"
```

Рис. 3.3.3 EditText занял всю ширину от guideline_start до guideline_end

4. Вывод

В ходе данной лабораторной работы были изучены основы верстки android-приложений и возможности и свойства таких контейнеров как LinearLayout и ConstraintLayout.

ConstraintLayout – очень мощный инструмент, но использование его во всех случаях – излишне. Если есть возможность без вложенности (или может даже с минимальной) использовать базовые контейнеры (FrameLayout, LinearLayout, RelativeLayout), то лучше ConstraintLayout оставить для более сложных макетов.

В результате работы удалось создать макеты в соответствии с придуманным дизайном.