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

1. **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

1. **Ход работы**
   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).

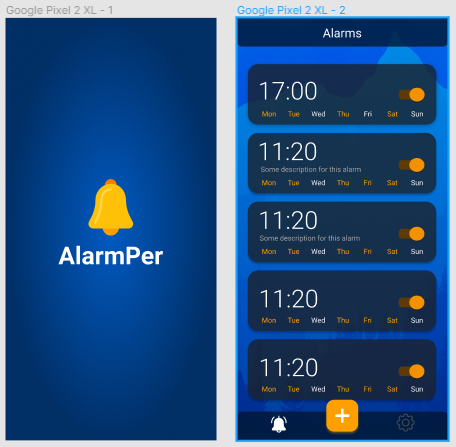


Рис. 3.1.1 SplashScreen и AlarmsList

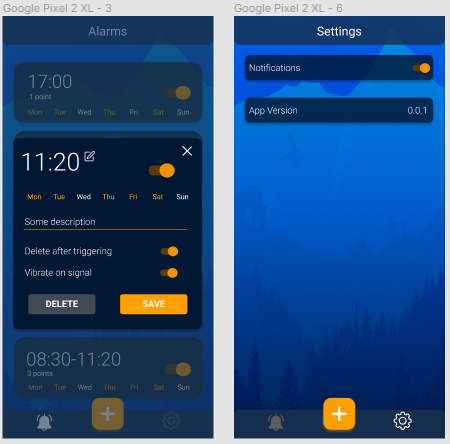


Рис. 3.1.2 AlarmDetails и Settings

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

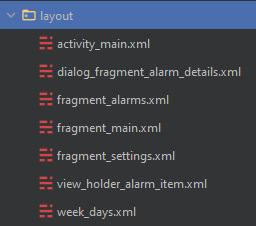


Рис. 3.1.3 Созданные XML файлы

Больше всего нас интересуют файлы dialog\_fragment\_alarm\_details.xml, где был использован ConstraintLayout, а также week\_days.xml, который в себе содержит LinearLayout.

* 1. **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>

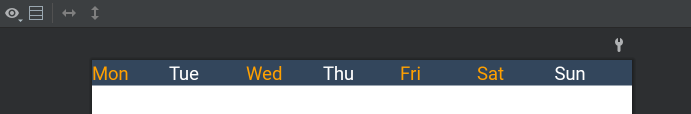


Рис. 3.2.1 Визуальный результат week\_days.xml

**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)

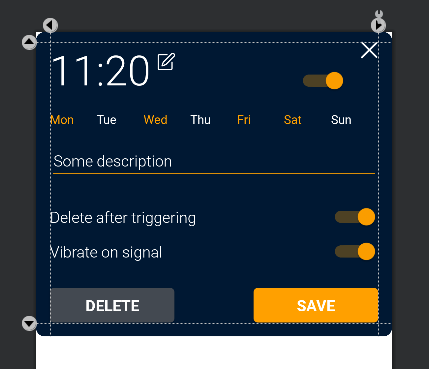


Рис. 3.3.1 Визуальный результат dialog\_fragment\_alarm\_details.xml

**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>

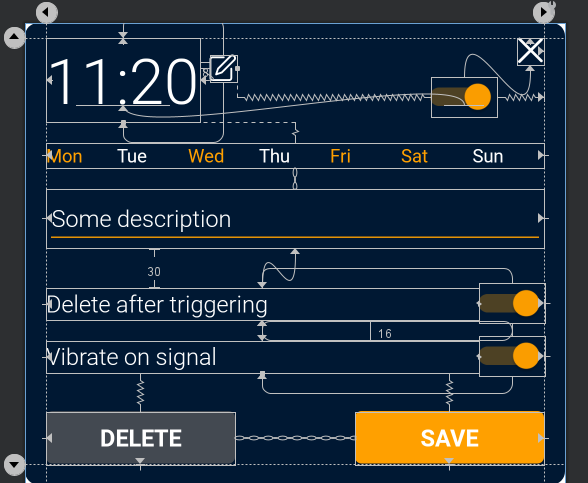


Рис. 3.3.2 Связи в dialog\_fragment\_alarm\_details.xml

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

1. **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.