Process:

**Layout:**

**Activity\_main.xml:**

Created different layouts:

1. **LinearLayout**: für die Anwendungshintergrund, die Hauptfenster. Es hat ihre eigene Eigenschaften und Merkmale
   1. **ImageView**: das ist das Anwenudngshintergrundbild. Sie wird bearbeitet um im Handybildschirm anpasst. Man kann auch Texte hinzufügen, Textgrößen bearbeiten usw..
   2. **LinearLayout**: es hat ein eigens id, “***idStartenBttn***” und eigene Eigenschaften und der Importsquelle. Es wird für den Button der Haupthintergrundsbild verwendent. Man hat jetzt ein Button aber es passiert nix, wenn man drauf klickt.
      1. **TextView**: es ist ein Zweit von b) und wird den Inhalt bzw. Kontext des Button enthalten, was erstellt worden ist.

**Uebungsaktivitaet.xml**

1. **Toolbar widget:** Addieren eines toolbar-widgets ganz oben auf der Seite bzw. Aktivität. Hier wird man einen Text sehen können als auch der Zurück-Button, dass zur letztverwendete Aktivität zurückbringt.
2. **LinearLayout:** Für das Aktivitätshintergrund und wird mit “idRestView” identifiziert. Sie hat wieder ihre eigene Eigenschaften und Merkmale.
   1. **TextView:** Text auf/innerhalb dem/des Button(s) gezeigt wird. Diese wird in der Übungsaktitivätshintergrund gezeigt und wird den Start der Übungen dienen. Es wird immer beim Endlauf der Zeit automatisch gedrückt, um die nächste Übung anzufangen. Der Text sagt: „Bereiten Sie sich vor“.
   2. **FrameLayout:** Die Grenzlinien der Untereseite. Es wird Dimensionen(dimens) und Farben(color) zugeteilt.
      1. **ProgessBar:** Der styl (**style)** des Fortschrittsbar wird ausgesucht. Es wird dann Eigenschaften zugewiesen und den Bar wird schließlich mit “idFortschrittBar” identifiziert werden.
      2. **LinearLayout:** Diese wird den Hintergrund des rundigen Button enthalten.
         1. **TextView:** Es hat die id “id***Zeitzaehler***”. Es hat die Farbe weiß und wird anfangen von 10 zu runterzuzählen. D.h. der eingegebene Text ist 10.

**drawable**

Hier sind alle die Bilder enthalten und deren Bearbeitung. Es wird auch den Form, Option, und Gestaltung der Bilder haben, was die verschiedene Layouts für den Zweck des Nutzens importieren werden.

**hintergrundfoto.jpg:** Das Bild was für die Hauptaktivitätshintergrund verwendet wird.

1. **item\_rundige\_colour\_accent\_border.xml:** Um zu definieren, welche Eigenschaften des Buttonsbox haben wird. Es wird erstmal al eine ovale Gestaltung definiert und die Margins werden die Farbe Grün übergegeben werden.
2. **item\_colour\_accent\_border\_ripple.xm**l:
3. **rundige\_fortschritt\_bar:**
4. **rundige\_fortschritt\_grau:**

**values**

1. **dimens**: Dimensionen, die in einem anderen Packet stehen, wo die LinearLayout, ImageView und TextView ihre Eigenschaften importieren können. Dies wird die Änderungsflexibilität erhöhen und wird dazu dienen, dass nicht den gesamten Code beeinträchtigt wird, wenn man sich zukünftig Änderung zu übernehmen.
2. **strings**: Die Namen der Bilder werden als Strings mit dem Text was dem Button geschrieben wird, definiert. Es wird dann in Layouts importiert usw.
3. **colors**: …

**fitnessApp**

**MainActivity**

1. import ***idStartenBttn***, which is the button we created in Layout

**UebungenActivity**

1. **Zwei globale Variablen für dem Ausruhen-Zeitabzähler:**
   1. **CountDownTimer: zeitUhrZaehler:** Eine Variable fuer Zaehler, die wir im Nachinhein instalizieren werden
   2. **restFortschritt:** es wird erstmal mit 0 instaliziert
2. **Functions:**
   1. **onCreate:** 
      * wir setzen zuerst ein Actionbar mit der Methode „setSupportActionBar“ in AppCompatActivity und wir geben unsere toolbar\_uebungs\_aktivitiaet in **layouts** als Parameter.
        1. Jetzt, checken wir ab, ob die actionBar gleich null ist, wenn nicht, dann sollte unsere actionBar immer aufgezeigt werden.
      * Jetzt erstellen wir eine ZurueckButton . Es gibt schon eine vordefinierte Methode, die das für uns machen kann. Sie heißt **onBackPressed().** 
        1. Es wird gelauscht, bis diese gedruckt wird und ein Event wird ausgeführt.
      * Wenn das alle am Lauf ist, dann können wir restViewEinrichten() aufrufen, um alles in der Aktivität zu zeigen.
   2. **setRestFortschrittBar:** 
      * idFortschrittBar hat eine Eigenschaft “progress” was zu übrigeFortschritt insalliziert wird.
      * Ein Objekt countDownTimer erstellen, die jede Sekunde einmal tickt. Dafür wird das folgende im Objekt gebraucht:
        1. **noTick** Funktion:
           1. beim jeden Fortschritt werden wir übrigeFortschritt installizieren und ihn mit 1 inkrementieren.
           2. Das idFortschrittsbar wird dann gleich zu restTimeDauer im Integer Form minus die übrigeFortschritt gesetzt.
           3. Das Timer wird jetzt im Textform (also, String) gezeigt.
        2. onFinish Funktion:
           1. wenn die Sekunden verlauft haben, dann beim Aktivität this, wird ein Text aufgezeigt.
   3. **restViewEinrichten:** 
      * Am Anfang wird den Audio abgespielt mit der Verwendung von MediaPlayer. Dann wird die idRestView sichtlich gemacht und den idÜbungsView versteckt.
      * wenn das timer/countDownZeituhrZaehler ungleich Null ist, dann canceln und der übrigeFortschritt gleich 0 setzen. If the timer is not running anymore
      * **Sonst,** setRestFortschrittsBar()
   4. **onDestroy():**
      * Wenn countDownZeituhrZaehler ungleich null ist, dann canceln und restFortschritt mit 0 setzen.
      * Sonst, super.onDestory(). Das ist eine Funktion in AppCompatActivity und sie wird die Aktivität beenden bzw. zuruecksetzen

**Model:**

**UeubungsModel:**

Es dient dazu, dass unsere Daten gepeichert bzw. verwaltet wird.