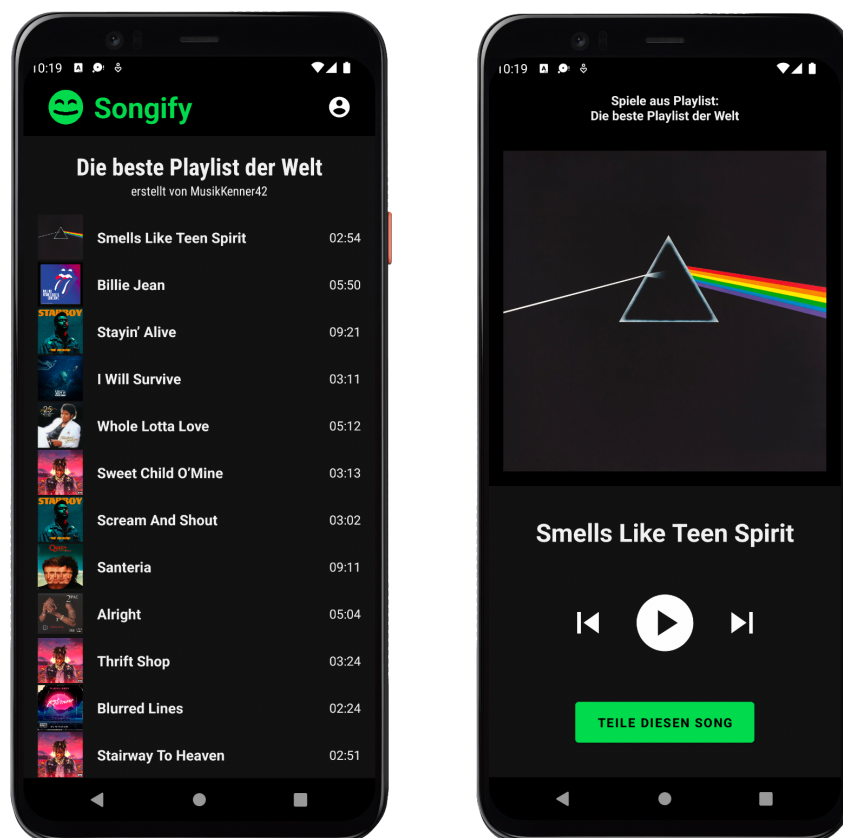


Hinweis: Zu bearbeiten sind Aufgaben 1 und 2. Aufgabe 3 ist eine Bonusaufgabe

1. MUSIK APP - Explicit Intents

In dieser Aufgabe ist eine `RecyclerView` mit einer Liste von Songs gegeben. Durch Klicken auf einen Listeneintrag soll eine Detailansicht aufgerufen werden, in der das Cover und der Songtitel zu sehen sind.



- Öffne das Projekt "Musik App", hier besteht bereits die bekannte Ordnerstruktur aus vorhergehenden Aufgaben
- zusätzlich zu den Layouts `activity_main.xml` und `list_item.xml`, findest du zusätzlich schon das Layout für die Detailansicht `detail_activity.xml`
- Los geht es mit der Klasse `ItemAdapter` der `RecyclerView`. Hier wollen wir die Listeneinträge klickbar machen und entscheiden, was bei einem Klick passiert
 - Füge in der inneren Klasse `ItemViewHolder` ein Attribut hinzu, welches auf die `CardView` verweist

- Hole die `CardView` in der Funktion `onBindViewHolder` aus dem übergebenen `ItemViewHolder` Objekt und speichere sie in einer Variablen ab
- Setze nun einen `onClickListener` auf die `CardView`, in dem die Detailansicht geöffnet wird
 - Erstelle innerhalb des `onClickListeners` ein `intent` Objekt, mit der Absicht, die `DetailActivity` zu starten

Hinweis: der Konstruktor für das `Intent` Objekt erhält als Parameter den übergebenen `context` und `DetailActivity::class.java`

 - Gib dem eben erstellten `intent` Objekt weitere Informationen auf den Weg, nutze dafür die Funktion `putExtra()` mit einem passenden Namen und dem entsprechenden Attribut als Parameter:
 - Das Attribut `imageResource` aus der `song` Variablen
 - Das Attribut `stringResource` aus der `song` Variablen
 - Starte die Detailansicht, indem du `context.startActivity()` aufrufst und das eben erstellte `intent` Objekt als Parameter übergibst- In der Klasse `DetailActivity`, in der Funktion `onCreate` sollen nun die übergebenen Informationen in das Layout geschrieben werden
 - Hole die `imageResource` und die `stringResource` aus dem `intent` und speichere sie in Variablen

Hinweis: Nutze dafür `intent.extras?.getInt()` mit dem entsprechend vergebenen Namen als Parameter

 - Teste, ob die Variablen `null` sind und wenn nicht, weise der `TextView` `tvTitleDetail` den Text zu, den du mithilfe der `stringResource` und `getString()`, bekommst.

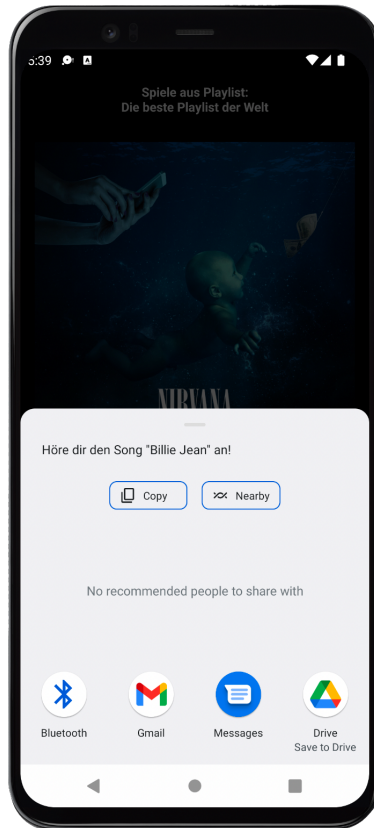
Weise außerdem der `ImageView` `ivCoverDetail` mithilfe der `imageResource` das entsprechende Bild zu
- Führe nun die App aus, bei einem Klick auf einen Listeneintrag sollte sich die Detailansicht öffnen und den entsprechenden Titel und das entsprechende Cover anzeigen!

Viel Erfolg!



2. SHARE - Implicit Intents

In dieser Aufgabe soll die Funktionalität des "Teile diesen Song" Buttons programmiert werden




- Programmiert wird die Funktionalität in der Funktion `onCreate` der Klasse `DetailActivity`
- Setze einen `onClick`Listener auf den Button `btnShare`
 - Erstelle innerhalb des `onClick`Listeners ein neues `Intent` Objekt mit der Absicht, etwas zu teilen

Hinweis: Übergebe dem Konstruktor des Intents dafür den Parameter `Intent.ACTION_SEND`

- Füge dem `Intent` ein Extra hinzu, mit den Parametern `Intent.EXTRA_TEXT` und dem Text, den du anzeigen lassen willst, z.B.: "Höre dir den Song "Billie Jean" an!"
- Gib dem `Intent` den Typ `"text/plain"`
- Erstelle das AuswahlObjekt mit `Intent.createChooser`, mit dem `Intent` als ersten Parameter und `null` als zweiten Parameter

- Starte nun noch die Activity mit `startActivity` und dem AuswahlObjekt als Parameter
- Führe die App aus, beim Klick auf den "Teile diesen Song" Button sollte sich eine Auswahl zum Teilen des Songs öffnen!

Viel Erfolg! 

3. ADVANCED - Next & Previous

In dieser Aufgabe soll die Funktionalität der “Nächster Song” und “Vorheriger Song” Buttons in der `DetailAnsicht` programmiert werden. Die App ist auf “shuffle” eingestellt, d.h. jedes Mal wird zufällig ein neuer Song angezeigt.

Hinweis: Es soll eine neue Activity erstellt werden, die die alte `DetailAnsicht` ersetzen soll.

Viel Erfolg!

