1 Permissions

Beim Nachforschen und bei der Implementierung wurden diese Quellen verwendet: [*Everything every Android Developer must know about new Android's Runtime Permission* [3]], [*Requesting Permissions at Run Time* [1]], [*System Permissions* [2]].

Permissions legen fest, auf welche Funktionen des Smartphones eine App Zugriff hat. Ziel dieses Systems ist es, Apps nur so viel zu erlauben, wie unbedingt nötig. Das bedeutet, dass es einer bösartigen App nicht möglich ist, erheblichen Schaden zu verursachen, ohne die entsprechenden Permissions zu haben. Da eine Applikation zu Beginn keine Permissions besitzt, müssen diese im Manifest deklariert werden. Android unterstützt mehrere Levels von Permissions.

- **Normal Permissions** stellen kaum Gefahr für die Privatsphäre des Benutzers oder den Betrieb des Systems dar und werden automatisch gegeben sobald sie im Manifest angefordert werden.
- **Dangerous Permissions** hingegen, können gefährlich für die Privatsphäre des Benutzers werden oder den Betrieb des Systems erheblich stören. Deshalb müssen Dangerous Permissions nicht nur im Manifest angefordert werden, sondern auch explizit vom Benutzer bestätigt werden.
- **Special Permissions** sind die dritte und seltenste Art von Permissions. Diese sind besonders heikel und müssen im Manifest deklariert und über einen Intent angefordert werden. Dieser Intent öffnet ein Fenster, speziell zur Verwaltung dieser Permission.
  Zu diesen Special Permissions gehören die WRITE_SETTINGS Permission, die Änderungen der Systemeinstellungen ermöglicht, und die SYSTEM_ALERT_WINDOW Permission, die es ermöglicht Fenster über allen anderen Apps anzuzeigen.

```
<manifest package="htl_leonding.fiplyteam.fiply"
    xmlns:android="http://schemas.android.com/apk/res/android" >
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WAKE_LOCK" />
  ...
</manifest>
```

Alle Arten von Permissions müssen in das Manifest eingetragen werden.

1.1 bis Android 5.1 (API level 22)

Wenn auf dem Gerät Android 5.1 (API level 22) oder niedriger installiert ist, werden alle Dangerous und Special Permissions auf der Google Play Store Seite der App angezeigt und müssen vor dem Herunterladen bestätigt werden. Sollten durch ein Update mehr Permissions benötigt werden, müssen diese beim Update bestätigt werden. Permissions können nur durch die Deinstallation der App zurückgenommen werden.

1.2 ab Android 6.0 (API level 23)

Wenn auf dem Gerät Android 6.0 (API level 23) oder höher installiert ist, wird beim Download aus dem Google Play Store keine Bestätigung der Permissions verlangt. Die Permissions werden nun während der Laufzeit der App abgefragt. Diese Abfragen muss der Entwickler selbst erstellen und anzeigen lassen.

```
final public int REQUEST_CODE_ASK_PERMISSIONS = 123;

public void CheckMusicPermissionAndReadMusic(Context context) {
  int readStoragePerm = ContextCompat.checkSelfPermission(this,
    Manifest.permission.READ_EXTERNAL_STORAGE);

  if (readStoragePerm != PackageManager.PERMISSION_GRANTED)
  {
    if (!ActivityCompat.shouldShowRequestPermissionRationale(
      this, Manifest.permission.READ_EXTERNAL_STORAGE))
    {
      showMessageOKCancel("Permissionmessage",
        new DialogInterface.OnClickListener() {

      @Override
      public void onClick(DialogInterface dialog, int which) {
        ActivityCompat.requestPermissions(Settings.this,
        new String[]{Manifest.permission.READ_EXTERNAL_STORAGE},
          REQUEST_CODE_ASK_PERMISSIONS);
      }
      }, this);
      return;
    }
  ActivityCompat.requestPermissions(this, new String[]{
    Manifest.permission.READ_EXTERNAL_STORAGE},
      REQUEST_CODE_ASK_PERMISSIONS);
    return;
  }
  rm.ReadSongsIntoArrayList(context);
}
```

```
private void showMessageOKCancel(String message, DialogInterface
  .OnClickListener okListener, Activity activity) {
  new AlertDialog.Builder(activity)
     .setMessage(message)
     .setPositiveButton("OK", okListener)
     .setNegativeButton("Cancel", null)
     .create()
     .show();
}
```

Da die Permissions erst während der Laufzeit abgefragt werden, ist es möglich nur manche von diesen zu bestätigen und so die Bereiche, in welche die App eingreifen kann, nach seinen eigenen Bedürfnissen zu kontrollieren. Zudem können Permissions jederzeit in den Systemeinstellungen des Geräts zurückgenommen werden.

1.3 Permission groups

Permissions werden in Permission groups zusammengefasst. Beim Anfordern einer Dangerous Permission, wird nicht die einzelne Permission, sondern die jeweilige Permission group angezeigt. Sowohl eine Anforderung der READ_CONTACTS Permission, als auch der WRITE_CONTACTS Permission, zeigt an, dass Zugriff auf die Kontakte des Gerätes benötigt wird. Wird eine Dangerous Permission angefordert, während die App schon eine Dangerous Permission derselben Gruppe besitzt, so wird die Anfrage automatisch bestätigt. Alle Permissions können einer Permission Group zugeordnet werden, allerdings sind nur Permission Groups von Dangerous Permissions relevant für Benutzer und Entwickler.