

北京理工大学计算机学院

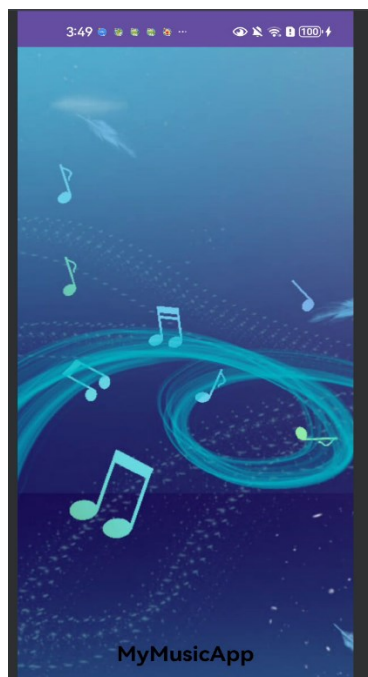
《Android 技术开发基础》课程设计模板

班级_____07122201_____学号_____1320221095_____姓名_____
徐帅_____

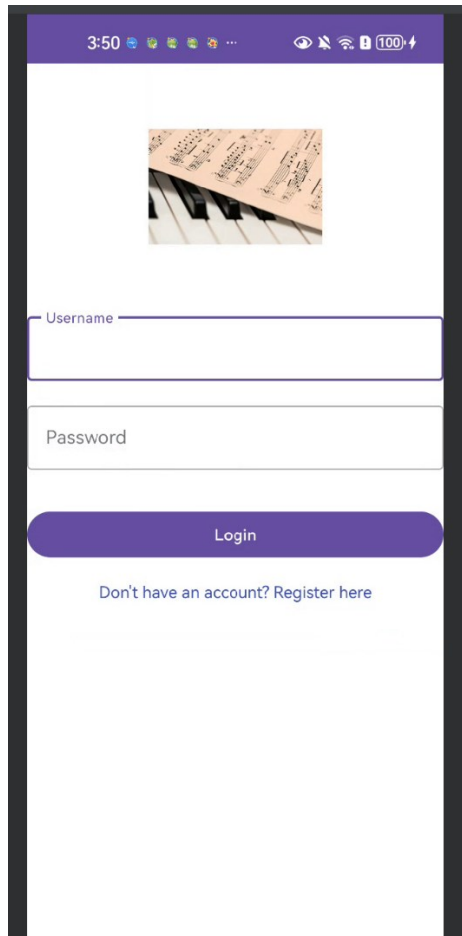
1 App 的运行与开发环境

- (1) **运行环境：** 用的是我自己的实体机，华为 nova 7 5G 版，最新的鸿蒙 4.0.0.116 版本。
- (2) **部署方法：** 直接安装 My Application.apk 即可
- (3) **开发环境：** Android Studio Iguana | 2023.2.1
- (4) **手写代码行数：** 约 2000 行（Kotlin）

2 App 功能说明：















开始页面，作为打开 app 的起始页面，两秒钟之后跳转进入登录页面。


A mobile application login screen with a purple header bar. The header bar contains a status bar at the top with the time 3:50, signal strength, Wi-Fi, and battery icons. Below the header bar is a white background. At the top of the white background is a square image of a piano keyboard with sheet music. Below the image are two input fields: the first is labeled 'Username' and the second is labeled 'Password'. Below the input fields is a purple button with the text 'Login'. Below the button is a link that says 'Don't have an account? Register here'.

登录页面，进行用户名和密码的输入，支持数字汉字字母等所有，不限制长度，登陆前需要注册。

3:53







Username

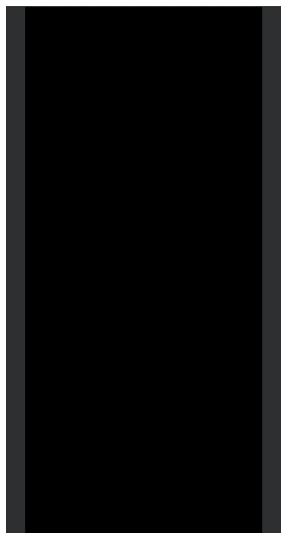
124379418kshjssb你好

Password

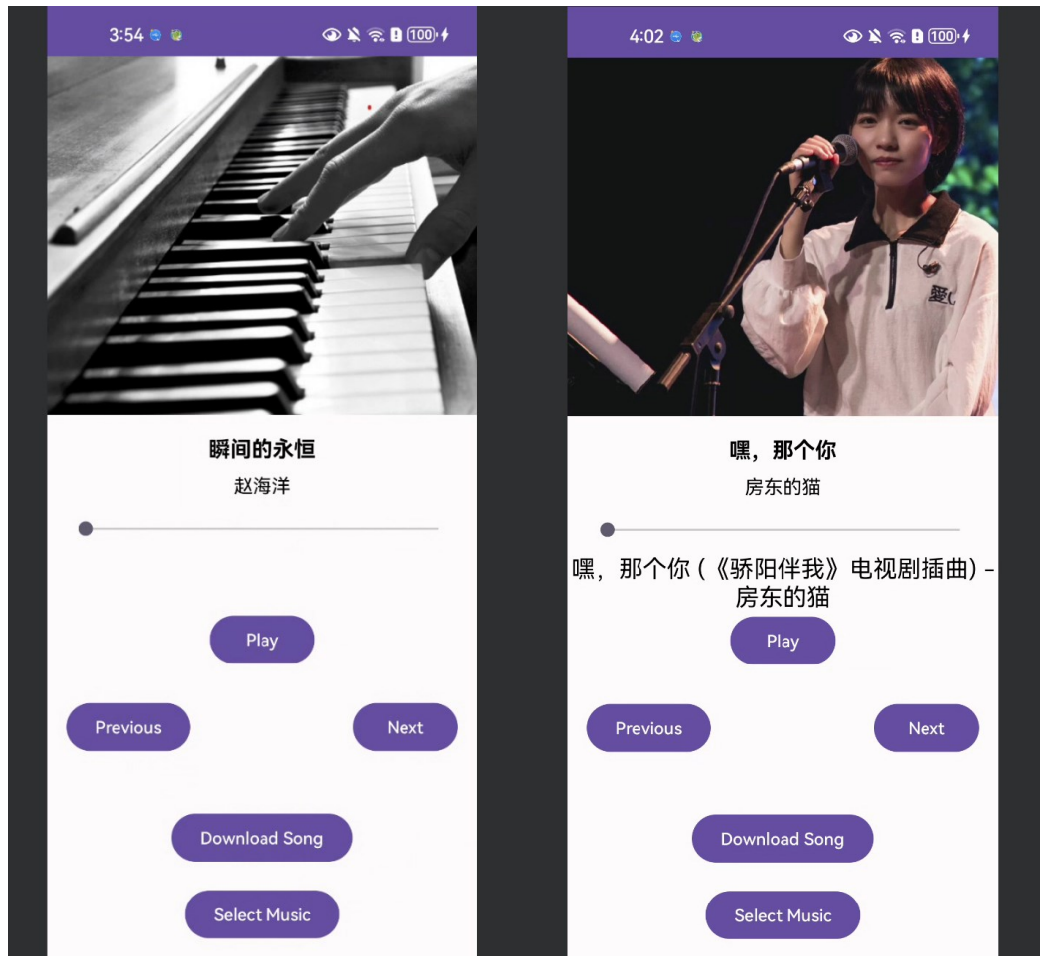
• • • • •

Register

Already have an account? Login here



注册页面，密码的输入有屏蔽功能在输入密码时只有手机端能看到页面，电脑的监控端会黑屏。



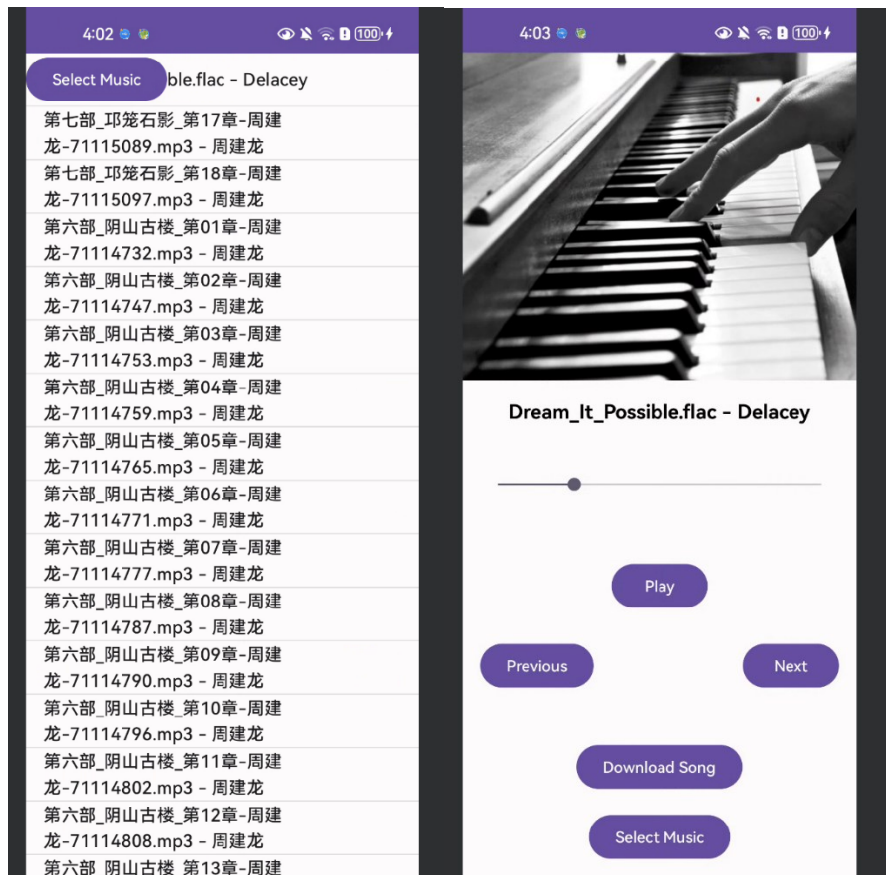
进入音乐播放器,有很多不同摁键我逐一解释。首先会显示这首歌曲的歌曲名字和作者名字,如果该首歌是纯音乐则没有歌词,如果不是纯音乐则会显示歌词。进度条可以随时拖拉,每首歌都会更新进度条,达到进度条与歌的进度和歌词进度完全重合。进度条播放完会自动进入下一首歌曲。

Play 摁键是开始和暂停。

Next 和 **previous** 分别是上一首歌和下一首歌。

Select music 的作用是点击后会扫描你手机内的所有 **MP3** 文件,这时需要你同意 **app** 使用手机内存。

点击 **download song** 会打开网页(默认打开百度),你可以在网页内下载歌曲到你的手机里。



这是点击 **Select music** 后，扫描我的手机之后生成的列表，想听哪个就点击即可播放。（我之前下载了好多有声小说）。

点击播放后就回到原来界面进行播放，按键和进度条都可以正常使用。



这是点击 **download song** 后跳转页面，你可以进行歌曲的搜索和下载。

我会将这个 **app** 的整体使用情况进行录屏（录屏收音不是很好），一起打包到压缩包里。

3 App 架构设计及技术实现方案

1. 加载音乐列表算法

算法步骤:查询媒体库获取音乐文件信息。遍历查询结果，提取音乐文件的标题、艺术家、持续时间等信息。将提取的信息存储到对应的数据结构中。

时间复杂度:查询媒体库: $O(n)$ ，提取信息和存储: $O(n)$

空间复杂度:存储音乐信息: $O(n)$

2. 加载歌词算法

算法步骤:读取歌词文件。逐行解析歌词内容，提取时间戳和歌词文本。将提取的时间戳和歌词文本存储到对应的数据结构中。

时间复杂度:逐行解析: $O(n)$

空间复杂度:存储歌词信息: $O(n)$

程序架构设计及技术实现方案

类关系

MainActivity: 主活动，负责播放音乐，显示歌曲信息和专辑封面，调整播放进度，选择音乐，下载歌曲等功能。

MusicSelectionActivity: 音乐选择活动，允许用户从设备中选择音乐文件。

Song: 歌曲数据类，用于存储歌曲信息。

Lyric: 歌词数据类，用于存储歌词信息。

包划分

com.example.myapplication: 根包。

activity: 包含活动类文件。

data: 包含数据模型类文件。

util: 包含辅助工具类文件。

第三方组件

使用了 **Android** 原生组件和支持库，如 **MediaPlayer**、**Uri** 等。

使用了 **Google Material** 组件，如 **TextInputLayout**、**Button** 等。

没有自定义组件。

UML 类图

	MainActivity

	- mediaPlayer: MediaPlayer
	- seekBar: SeekBar
	- playPauseButton: Button
	- prevButton: Button
	- nextButton: Button
	- songTitleTextView: TextView
	- artistTextView: TextView
	- musicImageView: ImageView
	- lyricTextView: TextView
	- lyrics: List<Lyric>
	- currentLyricIndex: Int
	- songs: MutableList<Song>
	- currentSongIndex: Int

	+ onCreate()
	+ initializeMediaPlayer()
	+ togglePlayPause()
	+ playPreviousSong()
	+ playNextSong()
	+ updateSongInfo()

```

| + updateSeekBar()          |
| + loadLyrics()             |
| + startLyricTimer()        |
| + updateLyric()            |
| + onDestroy()              |
| + onActivityResult()        |
-----

```

```

-----
|           Song              |
-----
| - title: String             |
| - artist: String            |
| - resourceId: Int           |
| - imageResId: Int           |
| - lyricResId: Int           |
-----
| + constructor                |
-----

```

```

-----
|           Lyric              |
-----
| - startTime: Long            |
| - content: String            |
-----
| + constructor                |
-----

```

包图

com.example.myapplication

```

├── activity
│   ├── MainActivity.kt
│   └── MusicSelectionActivity.kt
├── data
│   ├── Lyric.kt
│   └── Song.kt
└── util

```

组件图

MainActivity <--- MediaPlayer

MainActivity <--- SeekBar

MainActivity <--- Button

MainActivity <--- TextView

MainActivity <--- ImageView
MainActivity <--- MusicSelectionActivity
MainActivity <--- Song
MainActivity <--- Lyric

4 技术亮点、技术难点及其解决方案

技术亮点：

动态加载歌词：应用可以从歌曲文件中动态加载歌词，并且根据歌曲播放进度实时显示相应的歌词内容。音乐选择功能：提供了从手机储存中选择音乐文件的功能，使用户可以自由选择他们喜欢的音乐进行播放。歌曲下载功能：可以通过应用直接跳转至浏览器，方便用户下载他们喜欢的歌曲。播放控制：实现了基本的播放控制功能，包括播放、暂停、上一曲、下一曲以及进度条调整。

技术难点及解决方案

动态加载歌词：解析歌词文件并根据播放进度匹配对应的歌词内容是一个挑战。解决方案是编写歌词解析算法，并使用定时器实时更新歌词显示。音乐选择功能：获取设备中的音乐文件并进行选择需要考虑权限和兼容性问题。解决方案是在运行时动态请求读取存储权限，并使用媒体库查询音乐文件。歌曲下载功能：跨应用跳转至浏览器下载歌曲需要处理 `Intent` 的相关操作。解决方案是创建一个包含下载链接的 `Intent` 并启动浏览器应用。播放控制：管理 `MediaPlayer` 的播放状态和进度需要处理多线程同步等问题。解决方案是使用适当的锁机制和处理回调函数。

与其他类似的 App 相比的独特之处

应用能够实时显示歌曲的歌词， 提供了方便的音乐选择功能，用户可以从设备中自由选择喜欢的音乐文件进行播放。

我认为应用中最得意的地方是歌词显示功能。这个功能不仅提升了用户的听歌体验，还增加了应用的交互性和趣味性。实现这一功能需要克服一些技术难点，而最终的效果让我感到非常满意。

5 简要开发过程

5 月 6 号 设计了一个简单的音乐播放器，包括两个摁键，控制暂停和播放。

5 月 7 号 将两个摁键合二为一，并且新加入上下切歌的摁键，和导入了歌曲名字和歌手名字

5 月 12 号 导入了进度条，和图片，制作了登录和注册

5 月 13 号 制作了开始页面，进行了歌词的导入

5 月 14 号 学习了如何获取储存权限和网络权限，以及具体如何实现

5 月 16 号 完成了对自己手机内储存歌曲的使用

5 月 18 号 完成了上网下载歌曲和报告

6 学习感悟及对本课程的建议（可选）

我认为这是一节特别成功的课程，在我的大学学习中。通过对这门课程的学习，我真的

可以掌握到自己以后用得上的知识和技术，而不是为了应付考试和学分，我认为这个是很重要的。