

大作业要求

移动应用服务端要求

本次作业服务端不限制使用的编程语言、开发框架及采用的数据库，各小组依据成员技术熟练度以及项目要求使用恰当、流行的技术栈。

1. 网络通信 API 接口：服务端需提供 RESTful API 接口，支持与移动端应用进行 JSON 格式的数据交互。
2. 鉴权机制 用户注册与登录：服务端需实现用户注册和登录功能，并将用户信息持久化存储。接口鉴权：对特定接口实施鉴权机制，确保只有经过验证的用户才能访问。
3. 数据持久化 数据库选择：选择合适的关系型数据库或非关系型数据库进行数据存储。数据模型设计：设计合理的数据模型，支持基本的 CRUD（创建、读取、更新、删除）操作。数据安全性：对敏感数据（如用户密码）进行加密存储，采用适当的加密算法以保障安全。
4. 错误处理 错误响应格式：服务端需提供清晰的错误响应格式，以便客户端能够有效处理。日志记录：记录服务端的请求日志和错误日志，便于后续的调试和分析。

移动应用客户端要求

移动端使用 ArkTS 语言进行开发，小组在第一次作业的基础上完善应用功能，实现一个**完整的应用**（整个小组交付一份作业即可）。

1. 登陆注册功能

在 Lab 1 的基础上，实现登录注册功能，通过导航操作实现页面的切换与数据传递。如果登录失败，在页面下方弹出一个 Toast 提示框显示错误信息（使用 ArkUI 的 Toast 组件），并保持登录页面状态以使用户重新输入。

2. UI设计优化

实现页面的响应式布局，当设备屏幕方向改变时，音乐播放器控制栏和音乐列表区域的高度和宽度自适应调整，确保界面元素不会出现变形或遮挡现象。同时为页面切换和按钮点击等交互操作添加淡入淡出、平移等动画过渡效果，使界面交互更加流畅自然。

3. 多线程优化

了解ArkTS 中的线程池概念与用法，合理配置线程池参数实现多线程任务的高效调度与资源管理。线程间通信与数据共享机制，如使用消息队列、共享变量等方式在多线程环境下安全地传递数据与同步状态。多线程环境下的异常处理与资源回收机制，确保应用在复杂多线程操作中的稳定性与可靠性。 参考：

- [多线程并发-ArkTS并发-ArkTS（方舟编程语言）-应用框架](#)
- [业务模块并发加载场景-应用多线程开发实践案例-应用多线程开发-ArkTS并发-ArkTS（方舟编程语言）-应用框架](#)

4. 鸿蒙高级特性 - 分布式任务与跨设备交互（选做）

如果设备不支持可以跳过，选做有额外加分。

- 鸿蒙分布式任务调度的高级应用，包括任务拆分、分配策略与远程任务执行结果的回调处理。
- 软总线技术在跨设备数据传输中的深度应用，如大文件传输、实时数据同步等场景下的优化与可靠性保障。
- 分布式 UI 的状态管理与同步机制，确保多设备间界面显示的一致性与交互的协同性，实现分布式用户体验。
- 分布式设备发现与连接管理，包括设备搜索、配对、连接状态监测与自动重连等功能的实现，提升分布式应用的易用性与稳定性。

非功能要求

1. 代码协作 小组应自行选择代码仓库平台（推荐使用学校提供的Gitlab平台 git.nju.edu.cn 访问速度与稳定性良好），并利用 Git 进行协作开发，注意在大作业检查前不要公开仓库。
2. 文档记录 小组成员应通过文档简要介绍技术栈的选择（如为何使用NoSQL数据库），功能实现方式（如接口鉴权如何实现）。

评分

时间节点

2025.1.19 0点提交大作业。 19日安排线上检查。

交付要求

代码：提供代码仓库链接（邀请助教加入项目或在大作业截止日期后公开项目仓库） 文档：提供 项目介绍文档（介绍项目功能、交互方式等）、 开发文档（技术选型说明、接口说明文档等）、 过程记录文档（小组会议记录等） 演示：作业检查时现场演示项目，项目不需要部署，在本地运行并演示即可。

提交方式后续公布。

评分标准

//TODO 后续公布