1. 软件开发参考代码或链接或资料

[Android studio 自带虚拟机联网 - 哔哩哔哩](https://www.bilibili.com/opus/781505987323363349)

[Android中的Lottie动画入门 - 简书](https://www.jianshu.com/p/ddc22b094bd6)

[Android开发实战：利用系统自带日历控件实现高效日期管理 - 云原生实践](https://www.oryoy.com/news/android-kai-fa-shi-zhan-li-yong-xi-tong-zi-dai-ri-li-kong-jian-shi-xian-gao-xiao-ri-qi-guan-li.html)

[Android：Fragment addToBackStack()方法的使用-CSDN博客](https://blog.csdn.net/JMW1407/article/details/116561248)

[Android：Fragment详细介绍 & 使用方法解析\_android fragment-CSDN博客](https://blog.csdn.net/JMW1407/article/details/114069742)

[Android计时常用的7种方式\_android 计时-CSDN博客](https://blog.csdn.net/cpcpcp123/article/details/88545040)

[OkHttp网络请求框架在Android中的使用\_android 网络请求框架-CSDN博客](https://blog.csdn.net/IH_LZH/article/details/139087236)

[Android：安卓学习笔记之Service 的简单理解和使用\_简要解释一下service-CSDN博客](https://blog.csdn.net/JMW1407/article/details/122347723)

[Android权限管理及动态申请权限\_android 申请多组权限-CSDN博客](https://blog.csdn.net/weixin_56291477/article/details/122613092)

1. 本软件新增创新点介绍（包含本人开发部分的介绍，如已有理论课报告，直接写相对于理论课新增的功能即可）
2. 新增功能一：实现服务器与app数据同步

使用okhttp实现数据通信，在请求体中加入数据json，并在双端配置json解析器Gson实现数据通信。实现了单词学习记录，学习时长记录的上传服务器与同步到本地。功能需要用户登录账号，密码使用md5存储，登录后可同步对应账号数据。

1. 新增功能二：学习计时模块加入静音模式以及通知栏提示

静音模式：调用android勿打扰权限，启动静音模式

通知栏显示：启动计时后会在上方通知栏显示状态：目前时长/剩余时长，点击后跳转返回app。

同时由于静音模式需要“勿打扰”权限，通知栏也需要通知栏权限，后续还在welcome界面加入了权限检查以防程序抛出异常。

1. 突出创新点：

使用了request实现app与后端服务器的数据交互，权限调用实现静音模式，使用service实现了计时的通知栏交互

4）在新增各项功能中，本人完成了全部的开发。

1. 实验环境配置说明

1）代码运行需要的android库：  
 implementation("androidx.core:core-ktx:1.10.1")

implementation("androidx.appcompat:appcompat:1.6.1")

implementation("com.google.android.material:material:1.9.0")

implementation("androidx.constraintlayout:constraintlayout:2.1.4")

implementation("com.airbnb.android:lottie:3.4.2")

implementation("com.squareup.okhttp3:okhttp:4.12.0")

testImplementation("junit:junit:4.13.2")

androidTestImplementation("androidx.test.ext:junit:1.1.5")

androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

implementation("com.google.code.gson:gson:2.10.1")

implementation("com.squareup.okhttp3:okhttp:4.12.0")

implementation("com.fasterxml.jackson.core:jackson-core:2.15.2")

implementation("com.fasterxml.jackson.core:jackson-annotations:2.15.2")

implementation("com.fasterxml.jackson.core:jackson-databind:2.15.2")

完成包的导入运行即可，sqlite的模版db文件已在assets文件中存储，如apk未能正确导入db文件也会使用模版db文件复制一份文件。

2）后端配置：  
 运行框架：SpringBoot Version：3.1.2 （jdk：17）

数据库：mysql 5.7（需要提前创建好表结构）

依赖项：mybatisplus，hutool，knife4j swagger（用于测试）等

4. 功能验证：（软件操作说明+运行截图，避免大段代码展示）

1）功能一： 实现服务器与app数据同步

实现效果：

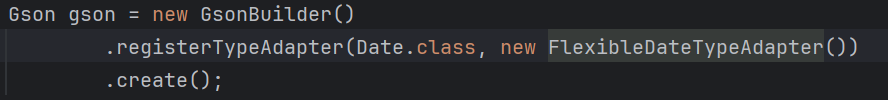
在用户完成登录后，在操作界面可以选择数据同步/上传数据来完成服务器与本地的数据交互，可以将本地数据上传至部署的服务器或是将服务器数据覆盖到本地：



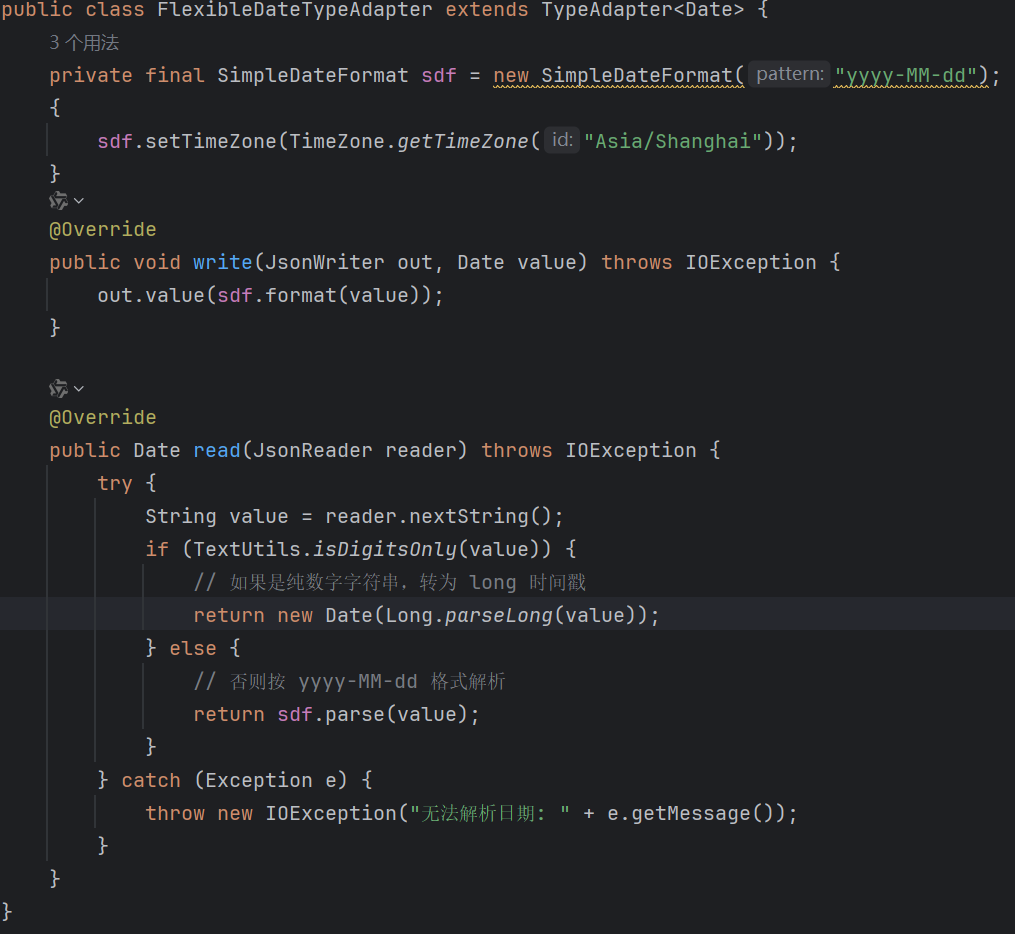
实现代码：

这里以数据上传为例，展示如何实现okhttp的request数据传输：

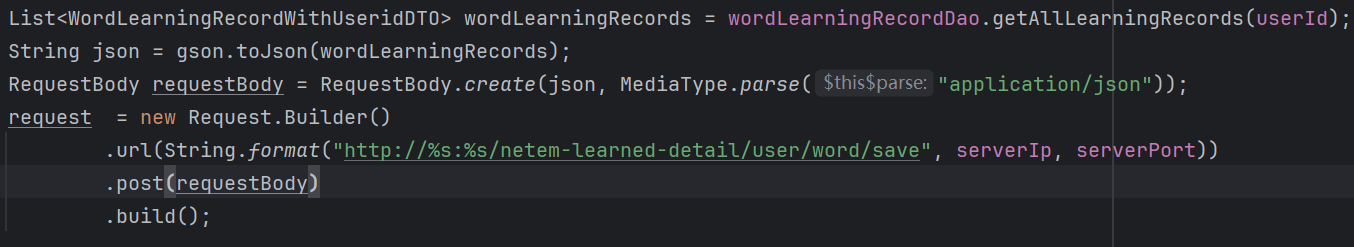
首先配置一个Gson的json转换器：



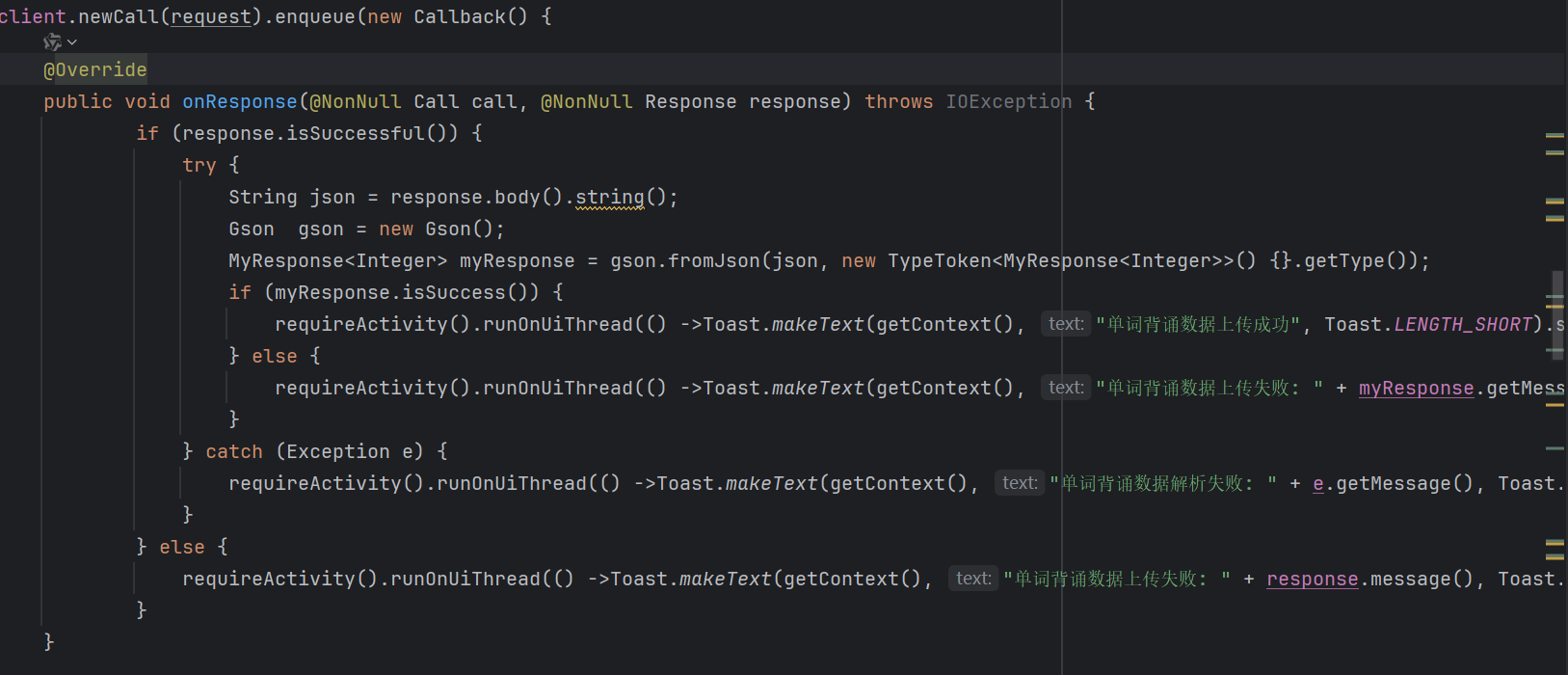
（sqlite输出的date可能会是时间戳，而在后端存储的date形式是YYYY-MM-DD，故需要手动配置一个adapter；同时时区也需要注意，即便MySQL的date设置过时区，双端的Gson或fastjson转换时也需要配置好时区问题，防止出现意外）



然后我们使用Dao提取本地sqlite数据，并使用Gson完成实体类的对象到json字符串的序列化，然后将之封装到请求体中，做成request：



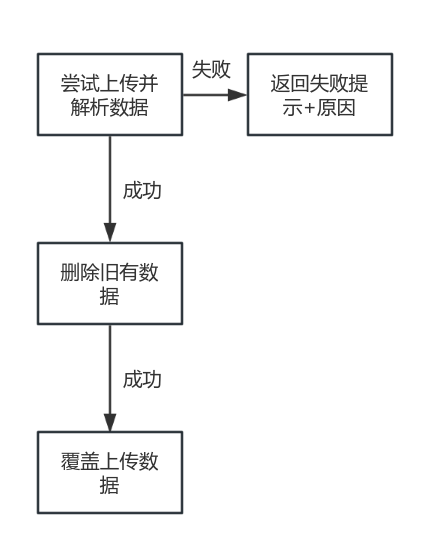
最后我们使用client发送该request，根据返回的response判断操作是否成功，并输出提示信息：



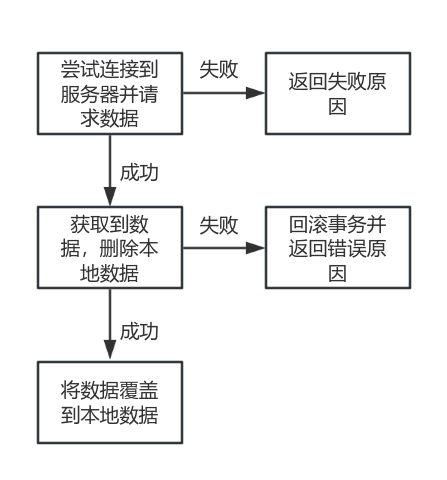
（其中response封装了两次是因为client以及SpringBoot的服务器会自己做一个response判断是否成功接收，而后面的response是我自己做的一个响应体，用于判断数据是否存在错误，解析是否成功等）

由此，我们就实现了对数据的交互：

数据上传：



数据同步：

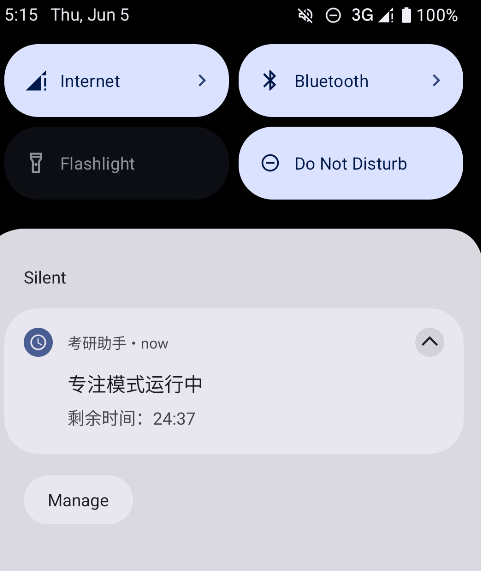


2）功能二： 学习计时模块加入静音模式以及通知栏提示

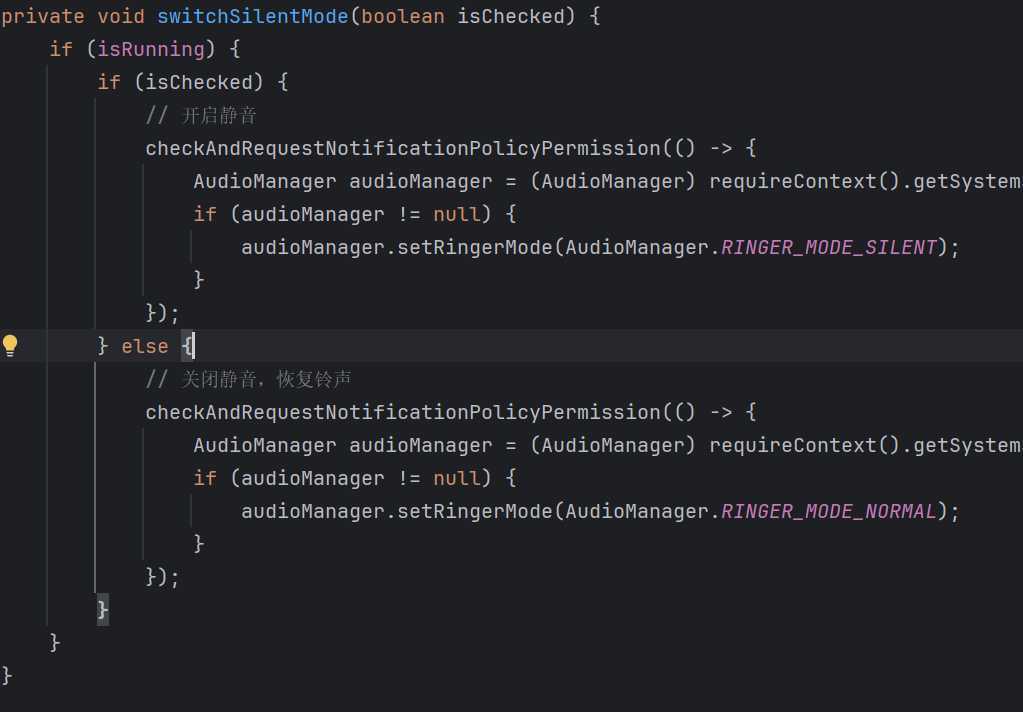
通过权限调用实现了静音模式的效果。通过service实现了通知栏显示时间与交互的功能：使用了通知通道（Notification Channel），适配 Android 8.0 及以上版本。

静音模式：

用户可以通过静音模式的滑动按钮实现静音模式的开启与关闭，当用户点击滑块按钮时会检测是否处于计时中，若为计时则会开启/关闭手机的“勿打扰”模式；当用户点击开始计时的时候会检测滑动按钮是否为开启模式，若为开启则会开启手机的“勿打扰”模式。

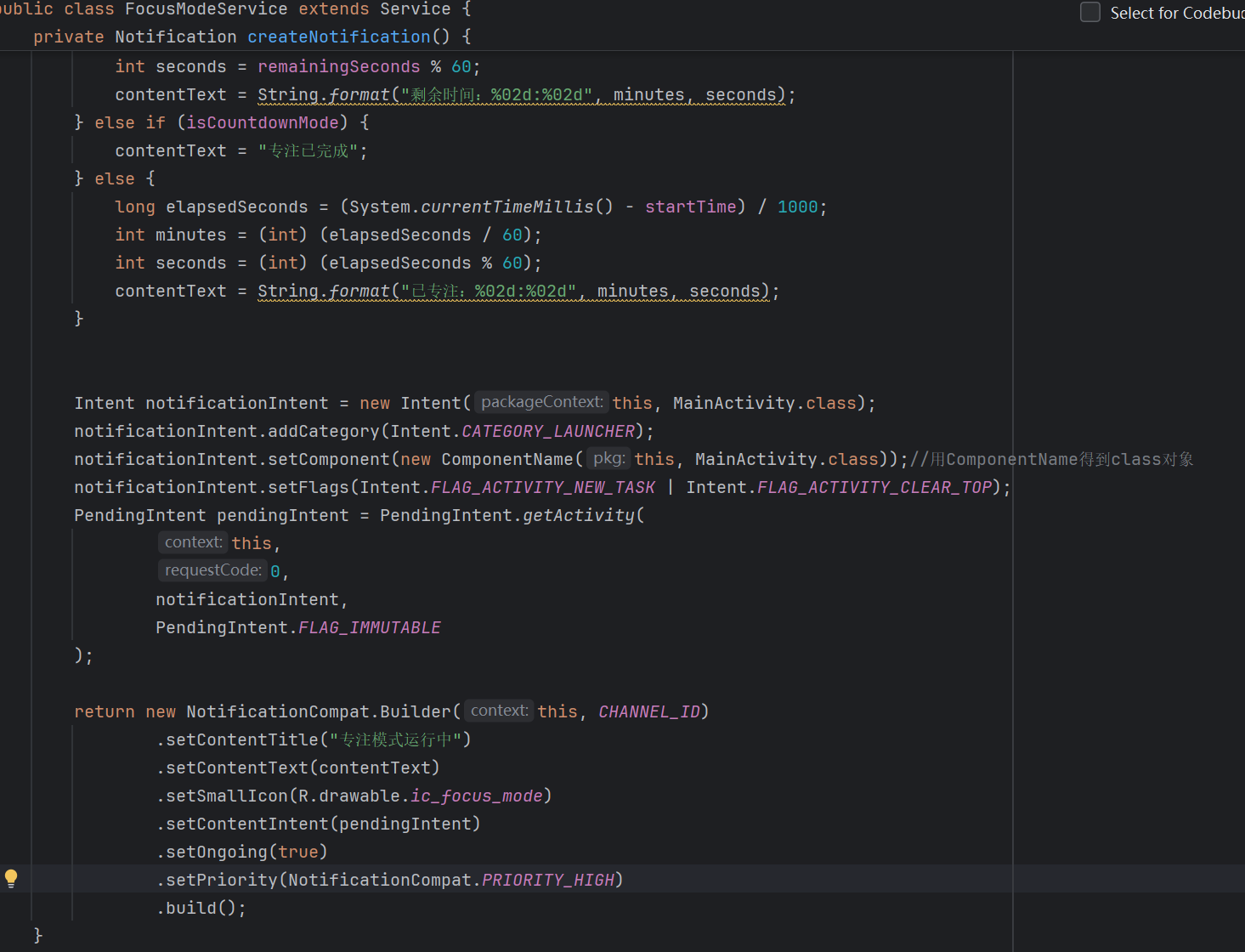


代码实现：



通知栏交互：

通过notification channel实现通知栏展示：



其中点击事件通过将mainactivity设置为singletask实现单例运行，并使用了

notificationIntent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK |

Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);

来实现点击时跳转到原先的app中（否则会导致生成一个新的进程，丢失原来的计时信息）

详细接口见“接口api说明.md”文件

3）突出创新点：

使用了request实现app与后端服务器的数据交互，权限调用实现静音模式，使用service实现了计时的通知栏交互

4. 个人在开发过程中遇到的问题及未来还需要增加的功能

遇到的问题：

在实现服务器与App数据同步时，遇到了数据格式不一致的问题。由于SQLite输出的日期格式与后端MySQL数据库中的日期格式不同，需要手动配置适配器来处理这种差异。此外，时区设置也是一个需要注意的地方，以确保双端的时间数据能够正确解析。

在实现学习计时模块的静音模式和通知栏提示功能时，权限管理成为了一个关键点。为了确保应用能够在用户设备上正确运行静音模式和显示通知，必须检查并请求必要的权限（如勿打扰权限和通知权限）。这要求在应用启动时进行权限检测，并根据用户的授权状态调整功能可用性。

在代码调试阶段，发现了一些逻辑错误和界面交互问题。例如，当用户尝试在没有网络连接的情况下上传数据时，如何优雅地处理这种情况而不导致程序崩溃或用户体验下降是一个挑战。通过添加适当的异常处理机制和用户提示信息，这些问题得到了解决。

在处理request请求时应当将请求部分写到service中来降低耦合度，serivice运行在后台，同时还可以不受Activity生命周期影响且不会影响主线程或ui线程的运行（不会导致未响应的发生）。

需要增加的功能：

个性化学习计划：基于用户的学习进度和偏好，提供定制化的学习计划推荐。

社交功能：允许用户之间分享学习心得、交流经验，甚至是一起组队学习，提高学习的积极性和互动性。

增强的数据分析：利用大数据技术，对用户的学习数据进行深度分析，为用户提供更加精准的学习建议和反馈。

语音助手集成：集成功能强大的语音助手，让用户可以通过语音指令操作软件，如查询单词、记录学习时间等，使操作更加便捷。

JWT自动登录验证：在后端设计JWT令牌，实现app自动登录的验证，保证用户账号信息的安全。