新系统：

一，架构与框架的必要性：

1 易于开发

1 框架会封装很多通用的方法，无需重复造轮子，util类，泛型类

2 框架提供很多机制（事务，拦截器，注入，安全，验证）

3 方便新手开发，框架具有封装性，无需了解底层运行，只要使用就可以了

4 易于调试，框架会提示详细的错误信息，方便定位和修复

2 易于维护

1 容易添加新的模块

2 删除，修改一个扩展模块不会影响其他模块的功能

3 每个模块负责一类功能，每个功能有且只有一个入口

4 接口化编程，替换新的文件时，只要使用相同的接口就可以保证正常服务

5 分层开发，每一层只负责一个业务阶段（视图，业务，储存）

二，选择什么框架，架构有什么要求

设计程序时有两个方向，一是吧程序做的足够严谨和复杂，保证每一个细节不出错，二是吧程序做的足够简单和易懂，减少出错的机会。使用框架可以显著减少代码量，框架提供的许多接口也可以限制程序运行的逻辑，每一步该做什么事，这样的好处就是能将冗长连续的逻辑分解为多个目的明确且简短的代码片段，减少出错的机会，即使出错也可以快速定位到问题的所在，。

我们选择框架是很重要的一点是，足够简单。简单，快速带来的好处就是你不需要很高的学习成本，也不需要很长的开发时间

第二点是有持续的维护和庞大的用户量，这样可以保证框架的安全性，和稳定性，遇到问题也可以有很多资料。

第三点是效率和泛用性，框架速度有快有慢，而且一个项目并不是一成不变的，有事需要考虑对项目的迁移和改造。一个团队对于一个泛用和全能的框架的熟练掌握也是一种缩短开发周期的办法。

什么是解耦，什么是冗余：

数据统计模块 需要调用其他的模块的dao层

是否有必要重新再写一个courseDao？ 新的dao却不能调用老dao丰富查询的方法，新dao需要用的时候还需要重写，这样并不算解耦，而是强行拆分成两个业务。这样做有什么好处？可以按模块出项目，但是很可能增加了重复代码，不方便维护，两个同名的dao路径不同，也很容易导致新手的混淆 --只要在原来的service层同时注入新旧dao就可以了

分模块开发是为了解耦，而不是为了强行分割业务。--尽量不修改原来的框架，增加复用性

我们这个数据统计并不是对所有类的简单统计，而是对课程，成绩，实验的深度统计，即使将来做扩展仍然只是横向增加个别统计对象。

是否可以完全解耦，不能解耦的部分

锚点类型的数据统计完全解耦，但是无法对特定的数据操作。