**体系结构设计文档**

项目名称：学习辅助系统

1. 技术栈选择

前端Vue所有技术栈：vue 2.9.6 + vuex + vue-router + webpack

UI库：element-ui

网络请求：axios

后端技术栈： Python 3.6 + Django 2.1.4

数据库：SQLite3

1. 三层架构

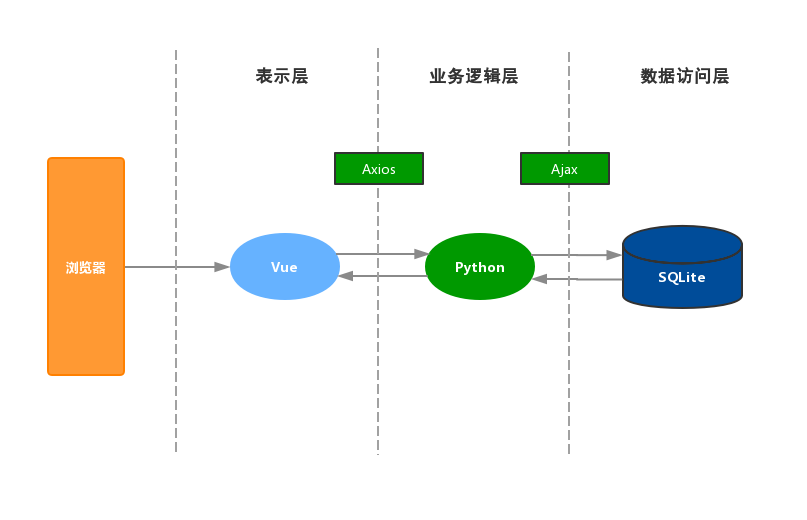
选择原因

三层架构(3-tier application) 通常意义上的三层架构就是将整个业务应用划分为：表示层（UI）、[业务逻辑层](https://www.baidu.com/s?wd=%E4%B8%9A%E5%8A%A1%E9%80%BB%E8%BE%91%E5%B1%82&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)（BLL）、[数据访问层](https://www.baidu.com/s?wd=%E6%95%B0%E6%8D%AE%E8%AE%BF%E9%97%AE%E5%B1%82&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao)（DAL）。项目采用三层架构设计，主要好处在于：开发人员可以只关注整个结构中的其中某一层；可以很容易的用新的实现来替换原有层次的实现；可以降低层与层之间的依赖；有利于标准化；利于各层逻辑的复用。

各层含义

1. 表示层：主要对用户的请求接受，以及数据的返回，为客户端提供应用程序的访问。

2. 业务逻辑层：主要负责对数据层的操作。也就是说把一些数据层的操作进行组合。

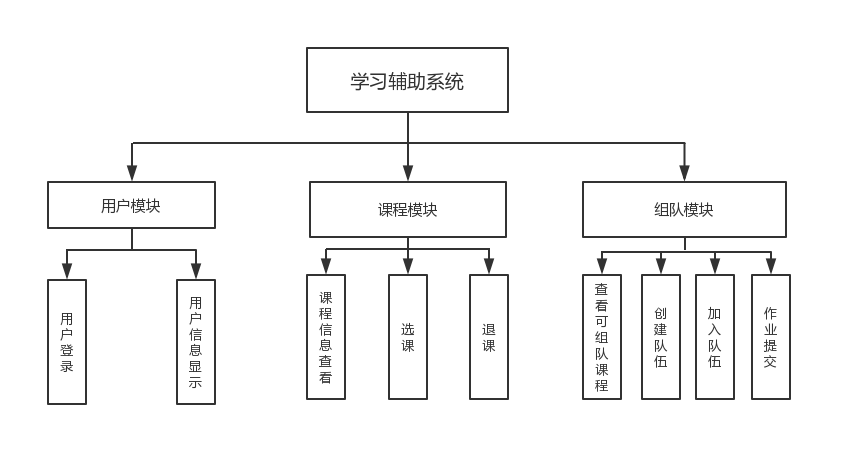
3. 数据访问层：主要看数据层里面有没有包含逻辑处理，实际上它的各个函数主要完成各个对数据文件的操作。

三层架构示意图

三、 系统结构设计

各模块功能描述：

1. 用户模块功能：
2. 实现用户注册及登录；
3. 实现用户信息的显示；
4. 课程模块的功能：
5. 实现课程信息的查看；
6. 实现选课；
7. 实现退课；
8. 组队模块的功能：
9. 实现可组队课程的查看；
10. 实现创建队伍；
11. 实现加入队伍；
12. 实现作业提交（word文档上传）



体系结构层次图