系统整理知识点，记录所碰到的问题，进行总结并且可以展示一些基本的代码学习尝试demo，作为事务管理系统，他应该可以管理文档和每日工作的记录，进行文档搜索和事务通知的功能。  
这个系统基于[https://github.com/xupingmao/xnote项目建立，修改增加新的功能，改变其页面展示同时作为学习python的一个途径](https://github.com/xupingmao/xnote%E9%A1%B9%E7%9B%AE%E5%BB%BA%E7%AB%8B%EF%BC%8C%E4%BF%AE%E6%94%B9%E5%A2%9E%E5%8A%A0%E6%96%B0%E7%9A%84%E5%8A%9F%E8%83%BD%EF%BC%8C%E6%94%B9%E5%8F%98%E5%85%B6%E9%A1%B5%E9%9D%A2%E5%B1%95%E7%A4%BA%E5%90%8C%E6%97%B6%E4%BD%9C%E4%B8%BA%E5%AD%A6%E4%B9%A0python%E7%9A%84%E4%B8%80%E4%B8%AA%E9%80%94%E5%BE%84)

工作的内容（不断添加）：

1. 登录系统中的用户信息及权限控制功能

2. 对于前后端约定的接口，采用Swagger框架生成模拟接口，并且生成接口文档,

此处应该在前端开发框架里面进行体现

3. 集成前端的mock服务，即前端可以在约定好接口的情况下独立的进行开发工作，这里的mock其实可以与swagger里面的接口文档进行融合

前端框架使用ext5.0 采用mvvm结构，后端采用python,减少工作中的学习成本，尽量使用已有的或可参照的系统进行开发，预计17年3月份能做出第一版的页面内容，前端的页面设计使用AdminLTE的模板进行开发，这样保证页面的可观看性和ui组件资源的有效性。  
对于第一期的工作目标，暂定为知识笔记，问题记录，工作总结，自测系统

知识管理系统：（第一期实现1,2,3,4四个功能）

1. 知识笔记：以博客方式记录分类各种知识。记得以标签进行分类

知识笔记里面应该有标签和分类，可以在线去添加新的知识，或者是进行摘抄和收藏，里面可以附加上思维扩展这一项，对涉及到的知识进行扩展。同时这个思维扩展可以添加到问题记录里面去，思维扩展的表现形式应该是qq对话框的形式，一问一答，然后可以针对性将每一组问答记入问题中去。

Node知识笔记部分后台实现的功能应该包括本地存放文件，和服务器端存放文件，并且可以实现同步，类似于git pull的功能，

* 可以选择本地创建好的文件，将其上传到服务器
* 可以在web端创建文件，其草稿保存到服务器
* 判断文件的名称是否一致，如果不一致则重新在服务器端保存一份

如果一致，则判断文件的修改时间，最新的文件覆盖掉老的文件

（实现增量更新的功能，即类似于git merge，在版本更新的时候不需要重新加载资源，只需要加载一段很小的 diff 信息，然后合并到当前资源上，类似 git merge 的效果）

* 同时也可以将服务器端的文件笔记存放于本地，文件下载功能

1. 问题记录：用问题单的方式来对自己困惑过的问题进行记录解答（分为曾处理，待处理，正关注，挂起中四个状态）
2. 工作总结：对自己生活和工作情况进行日报，半周报，周报，月报，季报等方式进行总结，类似于之前月报和日报的形式进行，具体可选但至少一周总结一次
3. 自测系统：一个简易的考试系统，进行自我测试
4. 任务提醒：以日历插件（calender）的方式同步或者建立关于知识点，问题记录，工作总结，自测系统等其他部分的任务或者提醒数据
5. 资料管理：记录保存学习资料里面各项资料的情况
6. 工具软件：记录更新各种工具软件的版本，使用方法，保存一份安装包
7. 系统功能：实现本系统的用户管理，自测系统测试资料上传，资料管理等数据
8. 代码实例：记录自己学习工作时搭建的各个框架，语言的代码demo，记录其安装环境，保证随时可以运行（与工具软件进行关联，保证可以及时搭建起demo代码的运行环境）
9. 数据分析：以数据图表的形式实时同步记录自己资料，知识点，问题，代码实例，工具，自我测试的情况

11. 轮子工厂，自己学习打造一个框架，可以分成多个组件，跟留下框架进行对比深入了解源码

对于登录页面，经过考虑，决定使用oAuth认证来做，页面除了基本的记住密码，注册，登录

一般情况下应该是访客直接登录系统，访客拥有最低的权限，可以查看知识笔记，问题单和其他的一个内容

管理员可以去删除其他用户的信息和数据，并且管理员用户具有访问任何一个用户页面的权限

一般用户登录的时候可以看到自己权限内部的内容，

登录页面的设计：左侧增加更新日志内容

