要求：

* 界面设计美观、简洁，不要太丑！
* 布局使用layui或者element的栅格布局
* 尽量全部的美化使用layui或element
* css和js使用太多，可以考虑独立为css或js文件，如果独立后造成问题，可以不独立
* 后台大部分的**逻辑实现**放在service层；
* controller层主要进行**数据接收返回**及**页面的转发和重定向**，功能实现调用service层的方法。

**周一：**

* **技术讨论**
* **完成简单功能的整体实现**
* **对复杂功能拆分，完成一部分实现**

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **任务** | | **负责人** |
| 1 | 登录 | 前后端 |  |
| 2 | 注册 | 前后端 |  |
| 3 | 登出 | 前后端 |  |
| 4 | 修改个人资料 | 前后端 |  |
| 5 | 考试入口 | 前后端 |  |
| 6 | 教师界面 | 前端 |  |
| 7 | 学生界面 | 前端 |  |
| 8 | 导航栏/首页 | 前端 |  |

* **注册：**

1. 能区分角色注册
2. 实现密码强度检测，密码太弱不允许注册
3. 实现验证码
4. 实现注册成功时发送激活链接（链接得形式讨论确定）
5. 邮箱已注册给予提示
6. 密码md5加密

* **登录：**

1. 能区分角色登录
2. 用邮箱进行登录
3. 邮箱未激活，不能登录，提示用户去激活邮箱
4. 登录成功，在session放入当前用户对象

* **登出：**

1. 登出摧毁session里的用户对象
2. 跳转首页

* **修改个人资料：**

1. 可修改昵称
2. 修改邮箱（讨论决定）
3. 可上传用户头像（头像图片命名可以用用户id）
4. 可修改密码，需要邮箱验证（验证方式讨论决定）

* **考试入口：**

1.邀请码可以以试卷id暂时代替，邀请码生成方式讨论决定