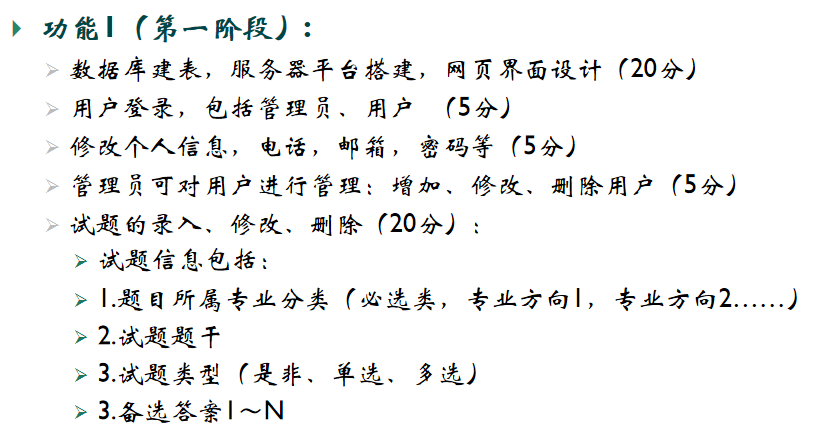
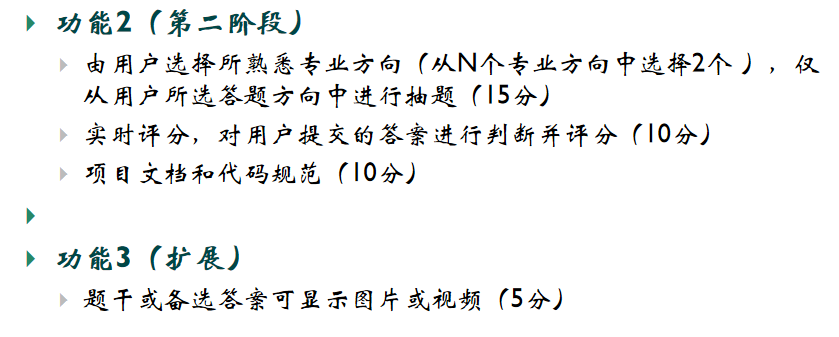
一、分工





具体工作大致如下

前端：个人建议是分配两个人，AB

使用vue开发

预期是

A当黑奴负责完成界面设计，完成登录界面，主界面，用户个人信息修改界面，管理员操作用户数据界面，考试界面等各自界面，也就是写html和css，只需要有一个空壳。

B负责完成空壳的功能，编写脚本决定什么时候向后端发送什么请求，再把响应结果做相应的处理显示到网页。

两个人把代码整合完build成dist文件即可

后端：三个人，CDE

使用springboot开发

C当黑奴，完成数据库的建表，插入足够量的数据供剩余的两个同学使用，由于工作量较少可能需要被抓去报告

D负责登录功能的开发，以及处理前端发来的用户数据修改与查询的请求，也就是说主要操纵用户表

E负责考试功能的开发，也就是主要操纵题库表和用户-题目关系表，

最后后端内容打包成.jar文件即可

二、工期安排

第三周周日之前各自完成对应的工作，第四周周四之前完成成员之间代码的整合，前后端的整合，最终的测试与调整

1. 接口文档

1.登录

（1）Url

测试时：localhost/login

实际使用时：服务器IP/login

（2）请求参数

①username（必填）

②password（必填）

（3）后端响应示例

{

"success":1,

"data":（令牌）

}

2.用户数据增加

（1）Url

测试时：localhost:81/accountInsert

实际使用时：服务器IP:81/accountInsert

（2）请求参数

①username（必填） //userid自动生成

②phone（选填）

③email（选填）

④password（必填）

（3）后端响应示例

{

"success":1, //1表成功，0表失败

"data":null

}

3.用户数据删除

（1）Url

测试时：localhost:81/accountDelete

实际使用时：服务器IP:81/accountDelete

（2）请求参数

①userid

②userid

.......

要求支持批量删除

（3）后端响应示例

{

"success":1, //1表成功，0表失败

"data":null

}

4.用户数据修改

（1）Url

测试时：localhost:81/accountUpdate

实际使用时：服务器IP:81/accountUpdate

（2）请求参数

①userid（必填） //userid不可更改，但需要填入要进行更改的的用户的id

②username（选填）

③phone（选填）

④email（选填）

⑤password（选填）

以上参数需要设置默认的初始值为原数据

（3）后端响应示例

{

"success":1, //1表成功，0表失败

"data":null

}

5.用户数据查询

（1）Url

测试时：localhost:81/accountSelect

实际使用时：服务器IP:81/accountSelect

（2）请求参数

①userid（选填）

②username（选填）

③phone（选填）

④email（选填）

⑤password（选填）

⑥admin (选填)

（3）后端响应示例

{

"success":1

"data":{

"userid":"114514",

"username":"mafty",

"phone":"1145141919",

"email":"1145141919@qq.com",

"password":"1145141919810",

"admin":1 //1表示管理员，0表示普通用户

}

}

6.试题增加

（1）Url

测试时：localhost:81/qInsert

实际使用时：服务器IP:81/qInsert

（2）请求参数

①qmajor（必填） //qid自动生成无需填写

②qtype（必填）

③qcontent（必填）

④qchoice（必填）

⑤qanswer（必填）

⑥img（选填） //这里上传图片给后端，后端接收存储到本地文件，之后前端的数据请求后端会返回本地文件地址给前端直接访问

（3）后端响应示例

{

"success":1, //1表成功，0表失败

"data":null

}

7.试题删除

（4）Url

测试时：localhost:81/qDelete

实际使用时：服务器IP:81/qDelete

（5）请求参数

①qid

②qid

.......

要求支持批量删除

（6）后端响应示例

{

"success":1, //1表成功，0表失败

"data":null

}

8.试题修改

（1）Url

测试时：localhost:81/qUpdate

实际使用时：服务器IP:81/qUpdate

（2）请求参数

①qid（必填） //qid不可更改，但需要填入要更改的题目的qid

②qmajor（选填）

③qtype（选填）

④qcontent（选填

⑤qchoice（必填）

⑥qanswer（选填）

⑦img（选填）

以上参数需要设置默认的初始值为原数据

（3）后端响应示例

{

"success":1, //1表成功，0表失败

"data":null

}

9.试题查询

（1）Url

测试时：localhost:81/qSelect

实际使用时：服务器IP:81/qSelect

（2）请求参数

①qid（选填）

②qmajor（选填）

③qtype（选填）

④qcontent（选填）

⑤qchoice（必填）

⑥qanswer（选填）

（3）后端响应示例

{

"success": 1,

"data": [

{

"qid": 1,

"qmajor": "math",

"qtype": 0,

"qcontent": "我是真的马夫蒂吗",

"qchoice": [

"是",

"不是"

],

"qanswer": "A",

"qimg": null,

"useranswer": null

},

{

"qid": 2,

"qmajor": "english",

"qtype": 1,

"qcontent": "原神",

"qchoice": [

"好",

"很好",

"非常好",

"特别好"

],

"qanswer": "D",

"qimg": null,

"useranswer": null

},

{

"qid": 5,

"qmajor": "english",

"qtype": 1,

"qcontent": "谁策划了9.11事件",

"qchoice": [

"孙笑川",

"陈睿",

"GoodEvening",

"哈基米"

],

"qanswer": "A",

"qimg": null,

"useranswer": null

}

]

}

//专业之间用"/"分割，0是非，1单选，2多选，单选就传单个字母比如"A"，如果是多选就传多个比如"ABC"，判断题传"A"表正确，传"B"表错误

10.查询是否有未完成的试卷（不加专业参数）

（1）Url

测试时：localhost:81/paperExistCheck

实际使用时：服务器IP:81/paperExistCheck

（2）请求参数

①userid

（3）后端响应示例

{ //存在未完成试卷的示例

"success": 1,

"data": [

{

"qid": 1,

"qmajor": "math",

"qtype": 0,

"qcontent": "我是真的马夫蒂吗",

"qchoice": [

"是",

"不是"

],

"qanswer": "A",

"qimg": null,

"useranswer": null

},

{

"qid": 2,

"qmajor": "english",

"qtype": 1,

"qcontent": "原神",

"qchoice": [

"好",

"很好",

"非常好",

"特别好"

],

"qanswer": "D",

"qimg": null,

"useranswer": null

},

{

"qid": 5,

"qmajor": "english",

"qtype": 1,

"qcontent": "谁策划了9.11事件",

"qchoice": [

"孙笑川",

"陈睿",

"GoodEvening",

"哈基米"

],

"qanswer": "A",

"qimg": null,

"useranswer": null

}

]

}

{ //此为没有的案例

"success": 0,

"data": "不存在已创建的试卷"

}

11.从头进入考试（加入专业参数）

（1）Url

测试时：localhost:81/testEnter

实际使用时：服务器IP:81/testEnter

（2）请求参数

①userid

②major

（3）后端响应示例

{

"success": 1,

"data": [

{

"qid": 1,

"qmajor": "math/chinese",

"qtype": 0,

"qcontent": "我是真的马夫蒂吗",

"qanswer": "A",

"qimg": null,

"useranswer": null

},

{

"qid": 3,

"qmajor": "computerScience/genshinImpact",

"qtype": 2,

"qcontent": "米哈游的游戏有A.鸣潮 B.王者荣耀 C.原神 D.绝区零",

"qanswer": "ABC",

"qimg": null,

"useranswer": null

},

{

"qid": 4,

"qmajor": "math/english",

"qtype": 0,

"qcontent": "你能分的清这是现实吗？A.能 B.不能",

"qanswer": "B",

"qimg": null,

"useranswer": null

}

]

}

12.用户作答保存（点击上一题/下一题/提交时对当页作答进行保存）

（7）Url

测试时：localhost:81/answerSave

实际使用时：服务器IP:81/answerSave

（8）请求参数

①rid

②useranswer

（9）后端响应示例

{

"success":1, //1表成功，0表失败

"data":null

}

13.试题提交

（1）Url

测试时：localhost:81/paperSubmit

实际使用时：服务器IP:81/paperSubmit

（2）请求参数

①userid

（3）后端响应示例

{

"success": 1,

"data": 33.3 //此为分数

}

四、前端注意事项

使用vue，两人需要事先协定好各组件的id再开工，由于组件繁多，交给前端的两位。

另外后端会将一些必要信息以cookie形式存放，供前端提取使用

主要界面有：

1. 登录界面
2. 主界面：开始考试（先选择两个专业），修改个人信息，修改试题库（仅在用户为管理员用户时进行渲染），修改学生信息（仅在用户为管理员用户时进行渲染）
3. 修改个人信息界面

对于普通用户只能修改自己的

对于管理员用户，罗列用户信息，分页还是滑轮以查看更多信息交给前端同学自由选择，要有输入筛选信息的组件，要有全选，全部删除的按钮，每个用户信息前要有选项框以供多数据删除

1. 修改试题库界面

大致同上

1. 作答界面

另外一个用户只能在同一时间作答一份试卷，主界面显示的时候需要先向后端发送查询用户-试题关系库的请求看看是否有该用户的数据，有则发送生成的请求无则直接载入试题

五、后端注意事项

* 1. 数据库相关注意事项

使用MySQL作为数据库管理系统

大致需要建表如下

用户（用户ID userid，用户名 username，电话 phome，邮箱 email，密码 password（不能明文），是否管理员 admin（1是0否））

题库（题目ID qid，所属专业（使用“/”拼接专业） qmajor，试题类型（0是非，1单选，2多选） qtype，试题题干 qcontent，选项数组qchoice,答案 qanswer，图片存储位置（没有图片就null） qimg）

用户-题目关系表（关系ID rid，用户ID userid，用户作答 usersanswer，题目ID qid，题目答案 qanswer）

(1)用户表事先操纵数据库录入或由管理员用户修改

(2)题库同上，另外图片按照url+服务器存储的形式，不放在数据库里，只要返回存储位置给前端让前端按照位置访问服务器中的图片即可

(3)用户-题目关系表在用户开始考试的时候插入相应元组，并将用户作答设置为null，在作答过程中修改用户作答这一字段为前端传输进来的用户答案，提交信息传入的时候按照用户ID选择该用户ID在表内的所有元组进行打分传回前端显示，之后删除相应元组。

* 1. 工程相关注意事项

密码要进行加密再送往数据库

前端登录后下发JWT令牌，并将一些需要的信息写入cookie

Springboot->maven构建工程，记得勾选mybatis，mysql，web的依赖

由于我们有对于动态sql的需求，就统一使用xml完成对数据库的映射而不是注解，这样修改会方便一些

controller组件中的方法记得要加上CrossOrigin注解，来解决跨域问题，类似下图，origin为前端的url，否则前端会无法正常得到数据

