# 进度文档

后台组 何松

项目 在线学习平台

## 写在开始的话

深思熟虑后, 还是选择了在线学习平台. 不因为别的, 其他的项目听起来就~~没那么熟悉~~. 换句话说, 与其挑一个自己感觉难做的去做一个抓人眼球的项目, 不如写一个烂大街一点的项目. 踏踏实实做好各个业务, 虽然在项目的新颖度上没有突出, 但是这四个题目做好了哪一个都是很厉害的了, 所以也不用奔着难做的去.

## 4月13日

今天正式着手准备!

首先考虑的问题就是**根据需求建表**

* **教师表**

基本用户类型之一, 考虑其登陆注册业务以及个人信息

* **学生表**

基本用户类型之一, 也考虑其登陆注册业务以及个人信息

* **盐表**

数据库安全的重要依赖因素, 每一位教师/学生都会有其唯一的盐值, 辅助密码的加密 添加教师/学生ID作为外键

* **课程表**

课程和老师挂钩, 故除了课程ID作为主键, 添加教师ID作为外键

* **选修课程表**

记录学生和其选修课程的关系表 添加学生ID和课程ID作为外键

* **章节表**

章节隶属于课程 添加课程ID作为外键

* **题目表**

题目隶属于章节 添加章节ID作为外键

* **学生答题记录表**

记录每个学生回答每一道题的信息 包括答题时间, 答案, 正确与否 添加学生ID和题目ID作为外键

* **学生章节学习情况表**

记录每个学生对于一个章节的学习情况, 包括答题数和正确率 添加学生ID 课程ID 章节ID 作为外键

* **讨论区表**

每一个课程有一个讨论区 添加课程ID作为外键

* **讨论回复表**

每一条回复与讨论区和回复者挂钩 添加讨论区ID和教室/学生ID为外键

十一张表... 我学JavaWeb就最多同时接触三张表... 有点难绷...

面对这么多表, 我有种难以言喻的不安, 总觉得某一张表建得不够好, 会在后面的开发中影响我. 但是说实话, 这已经是我现阶段分析需求之后能做到的最好了. 要是去担心开发之后才会知道的问题, 也不太现实, 纯属内耗. 故今天的建表工作就此.

## 4月14日

今天根据昨天建好的表, 再次分析了一下需求, **使用UML序列写了业务流程图**. UML序列这东西我是误打误撞发现的, 发现意外地适合写业务流程图.

打算明天就开始写登陆注册功能了. 着重解决令牌和密码前后端协同加密. 应该是一个大工程. ~~话说明天是满课的周一~~

## 4月15日

严格意义上现在是16号了...修bug修到半夜三点...

白天再补上15号的详细开发过程.

对于教师注册服务(教师和学生注册大差不差), 我思考了很久, 因为要考虑到加密的问题.

有几个前提:

1. 传输过程不能出现明文密码
2. 加密解密的密钥前后端需要统一

得出一个结论: 前端需要拿到后台生成的唯一密钥, 根据密钥加密后再进行密码的传输.

那么进入实际业务的流程: 前端在用户输入姓名电话密码准备注册时应该传输哪一个信息给后台**作为查询数据库是否是已存在用户的依据**呢? 一般不会在学习平台上要求注册时自行设置ID, 而且名字也有重复的可能性, 那么只有电话这一个依据了

故在前端先进行电话号码的传输, 后台接受到之后根据身份类型查找教师, 检查返回值是否为null, 若是则说明为新用户, 后台生成盐值转化为字符串之后传输到前端,前端又依靠接受到的盐值对密码加密, 随后将加密后密码还有盐值随着表单数据一起传输到后台, 保存到数据库.

那么学生注册也是根据这一套流程来进行注册.

明天打算补完登陆验证.

## 4月16日

今天大改了加密业务.

原本我打算是通过前端加密->后端解密这种模式进行登陆的验证, 后面发现我开始使用的加密方法由于各种个样的原因, 使用相同的原始密码和盐值加密和解密, 后端解密时会出现乱码... 搜索了很多资料都没有解决, 故只能**抛弃盐值表**, 在注册时, 通过账户重复性验证后前端直接使用SHA-256将明文密码加密后传输到后台进行存储, 每次登陆时需要在前端对用户输入的登陆密码加密后传输到后台, 后台根据ID和密码去查表, 查得到用户则登陆成功.从始至终传输过程没有出现明文密码, 至少实现了这个加密的功能. 不过这样子密码就只能通过邮箱和手机进行重置了. 也不是什么难事对吧?

然后将教师和学生的登陆界面设计完毕. 明天打算完善一下过滤器, 之后开始着手完成学习平台的相关业务.

## 4月17日---还剩十天

加密业务改完之后在登陆验证这一块就方便一些了.

临时补了Filter的相关知识, 算是将登陆注册和登陆验证这个坎跨过去了, 就是每次前端都需要加个请求头带着令牌, 有点烦. 然后也做出来了教师的首页, 并且完成了个人信息查看和修改的功能, 不过**各个表单的验证这一块还没做完善**, 仅仅在注册的表单上使用了前端的输入检测来控制输入合法性. 后面都需要补上, 而且**日志的记录也有欠缺**, 都是需要在后面~~补注解的时候~~去修改一下的. 说实话前端才是这次对于后台组的考验......因为了解太少了, 一直在埋头学后台的技术, 其实要是硬说, 前端也能写, 就是调试起来真的绝望, 而且代码可读性完全没有Java好. 这些导致找bug难上加难, 以至于今天一度崩溃...

明天打算写好教师创建课程, 编写题目, 以及学生的个人主页信息修改以及课程参加业务. (做完登陆注册验证业务说话就是硬气)

## 4月18日---还剩九天

写完了学生的个人主页信息修改, 以及教师的创建课程. 在思考和每个课程绑定的章节以及和每个章节绑定的题目应该展示在什么地方.

今天和学长抱怨了这几天写前端的烦躁, 虽然学长指出我策略有误不应该花时间在登陆验证上, 但是不论对错, 只要是一个人做前后端, 不管哪个模块都会遇到很大的问题的. 我还是觉得最终考核和训练营的差距太大了(指后台组的前端部分) , 只能去熬夜弥补...这不是用框架简化操作就可以忽视的, 真的不如减少一些大组的作业然后小组加上前端框架的学习, 这样起码能让我们在最终考核这种巨大难度的挑战上能有合适的知识储备来写. 只是个人的一些看法, 我还是很认真在写项目的...

晚上去实现学生课程参加业务时在后台尝试了连表查询, 发现了一个很不得了的东西, 原来连表查询不被推荐的原因是因为两个查询语句会产生两个resultSet, 这俩会互相影响导致后续报错说结果集已关闭. 故只能减小mapper的颗粒度, 一条一条写了.

又是踩坑的一晚, 本来想让学生只能选没选过的课程, 自制的mybatis功能又太简易, 有些地方赋值会有问题, 只能白痴一点在选择课程的时候无脑全部查询了(当然不会查到过期的课程).

追加完成了课程参加业务. 明天要补上添加章节还有题目的业务, 希望可以完成学生进入课程学习的业务.

## 4月19日---还剩八天---接近中期检查

今天有一个好的开始! 昨晚因为思考**学生挑选课程时应该如何实现只能挑选没参与过的课程** 而尝试了很多方法.

1. 就好像昨天说的连表查询, 虽然不知道是不是我说的两个结果集冲突的问题, 反正是不能实现这个要求
2. 然后我又想去改造一下mybatis, 用where courseID NOT IN (xxx)来筛选, 这个时候选择一条一条sql查询语句来编写. 然后我发现我查出来学生参与了的课程ID, 这些课程ID还需要我获取一个connection对象, 调用它的createArray方法把这些课程ID转化成一个Array类型的数组来给这个xxx赋值, 可行, 但是又有bug, 说是SQLFeatureNotSupportedException , 查了一个说是驱动不支持, 我可不敢改版本了, 谁知道又给我抛什么新的异常.
3. 然后我选择不去对sql处理, 我直接去对sql查询后的结果进行处理.
   1. 先查询在有效时间内的课程courses
   2. 再查出学生参与过的课程
   3. 获取学生参与过的课程IDenrolledCourseIDs , 使用迭代器去对courses 处理, 找到课程ID包含在enrolledCourseIDs里面的就remove掉. 迭代器支持在遍历的时候对List进行增删. 然后获取到的处理后的courses 只包含学生没参与过并且在时限的课程. 直接返回到前端, 学生打开课程选择页面时只能看到他没参加过并且有效的课程.

* 麻烦了一点, 但是实现了业务就皆大欢喜!
* 后面要去实现章节和题目的添加以及学生的参与课程进行学习.

晚上做完了添加课程章节的业务, 不过还是需要思考添加了课程之后应该怎么安排添加题目的业务, 应该如何展示出来.

还有, 题目表也根据不同类型细分了表, pojo也相应地变化了一下.

题目表由原先的一张表**细分为四张表** , 一张总表记录每个题目的共性如题目ID章节ID以及题目内容和类型.

然后根据题目类型新增了三张表: **判断题表** **简答题表** **选择题表** 根据实际类型记录相应信息.

四张表对应pojo类也编写了, 不过dao层的操作到时候应该会比较麻烦...啊啊啊不想再修bug了!!!我好想念mybatis, 自己手搓的真就是垃圾啊...

明天要写完题目添加和数据回显, 最好能做到学生进入课程学习.

## 4月20日---还剩七天---明天中期考核

对问题相关的表的一些结构做了修改, 后台使用多态的写法来整体地查询章节的题目.

1. 表结构引出的问题

原本根据超类Question 单独建了一张表, 为其他子类单独建了三张表各自存储, 后面结合手写mybatis对表进行curd时发现有点困难.

* 每新建一道题目, 传输到后台借助json序列化工具转化为javabean, 这个时候如果想存储题目信息进表里面, 首先要取出题目共性存到超类的表, 再根据题目各自特性存到各自的表. 这个其实实现起来不难.
* **难点来了** , 如何取出这个题目信息? 我自己手写的mybatis毕竟有局限性, 执行查询操作时, 严格要求返回结果里指定的pojo (或者是单个实例, 或者是实例的集合)的每一个成员属性在表里头都必须有对应的字段, 否则就会出现异常.这个时候我一开始的分表策略就不管用了. 存储题目特性的表根本没有对应的pojo.

1. 解决方法

那么在这个局限性下, 我只能修改分表策略, 干脆三种题目三种表, 想查询哪种题目就去哪一张表进行查询, 这样至少实现了题目数据的查询, 只要拿到章节ID就能查到其下的题目. 当需要查询章节的所有题目时, 由于细分后题目是题目的子类, 只需要**根据章节ID在三张表依次查询, 将得到的三个List List<TrueFalseQuestion> List<ShortAnswerQuestion> List<MultipleChoiceQuestion> 全部加进一个List<Question> 里面, 将其转化为json之后**传输到前端就行. 而修改操作则按照题目类型去对应的表进行操作就行了.

说实话题目这一块我思考甚久, 我一度认为题目的curd是这个项目的最大难点, 但是当我想出**多态** 之后问题就迎刃而解了. 谁明白当我看到List<Question> 转化成json后每一个元素都没有丢失其属性的时候我有多开心! 因为之前我尝试前端传输题目集合到后台进行处理, 我将从请求获取到的json数据根据Question.class 进行转换之后发现它有字段丢失的问题.

前端对于题目类型根据题目数量 动态展示这一块也耗费了我很长时间, 不过兜兜转转还是通过组件的属性传递解决了, 这里不展开讲前端的开发过程. 总之是完成了教师这一块对于课程, 章节, 题目 这三个互相依附的东西的增加和查询. 删除和修改还是好做的, 后面完善的时候再做.

明天要去中期检查了, 希望能有个好结果, 当然进度也要跟上, 明天尝试完成学生进入课程学习和做题的业务.

## 4月23日

前两天没写进度文档, 不过也没落下进度, 只是修修补补.

中期检查没被怎么给建议, 就是几个比较小的建议, 感觉我做出来的东西还是挺正常的.

前两天也是完成了学生做题, 还有学习章节数据回显的业务, 不过有一点比较难实现的就是学生做题记录了. 后台查询这个记录倒是不难, 但是前端如何展示这个记录又是一个难题, 故只做到了每次提交章节作业之后就锁定提交按钮并且显示正确率. 不想在前端浪费太多时间了, 还有教师的课程信息监控呢...

于是今天打算完成教师的课程信息的查看.

查看参与学生的业务完成, 但是查看学习情况就稍微有点棘手了. 明天想想方法解决, 然后准备搭建每个课程下的讨论区.

题外话, 距离新建文件夹已经过去十天了, 这十天我现在想想, 真不知道我怎么熬过来的. 后台要融合我自己手写的mybatis还有连接池, 前端要使用没有系统学习过的Vue+ElementUI, 从零开始打造一个这么综合的项目...... 当时看着这个题目审题的时候只是觉得, 关系有点复杂呢, 但是实际写起来, 真牛魔的要命, 连续熬夜六晚到三点不是瞎熬的... 期间后台的手写mybatis又出了不少问题, 终究是简易版的手写框架, 处处都是局限性, 单单花时间找bug就是一晚上, 然后前端更不必说, 出了bug都不知道如何调试, 真的是绝绝又望望. 不过都熬过去了, 我做到了. 这就足够了, 足够对寒假前立志进QG的我说一声,

**我辛苦了. 继续努力吧.**

## 4月26日--结束

熬到了...

我其实还有几个有点重要的业务没做, 不过也只是在前端的局限而已, 后台的思路我还是有的.