

个人总结

徐奕帆 171250007

担任组长从头开始一个项目对我来说是不小的压力，并且是在这样比较特殊的时期，不能面对面地开例会，沟通有一点点的不便利，但总体上来说我和我的队友们还是尽自己所能克服一切困难，努力地完成了迭代一的任务。

作为组长，这一次我有很多不足的地方，最首要的就是时间安排，因为部署 Jenkins 的问题，我迟迟没有开始项目启动，文档的时间安排也比较松散，直到开始代码工作才发觉时间真的非常紧急，非常感谢我的组员们愿意陪我一起极限操作，这一点在接下来的阶段中一定要好好改正。另外我负责的是前端、测试、还有代码整合工作，这些工作对我来说都是不小的挑战，这一次的任务给了我很多经验值。

迭代一完成得还算差强人意，这个项目接下来的工作我相信我们小组会以更好的面貌来面对挑战。我们一定会吸取经验教训，在接下来更合理地安排工作、完成任务。

赵茁羽 171250006

我这次主要负责前端和代码测试部分，因为软工二负责的是后端，所以这次想试试前端。一开始本来想写前后端分离，但是迭代一的 ddl 有点紧，而且组内没有成员有经验怕到时候出了难以结局的问题会来不及，最后还是保守起见参照软工二构建的基本框架。

这次是完全线上的讨论，感觉有利有弊吧，利是大家交流多了，有想法就会说，弊是无意义的交流会比较多，容易讲着讲着就扯了。不过对写代码说在家是挺舒服的，一写起来浑天黑地，没有外界干扰。

第一次从零开始做一个完整地系统还是蛮考验人的，我们组没有大腿抱，大家都是边学边写，有 bug 互相帮忙，氛围很好。感觉这次最大的问题是时间分配不好，刚开始在家上学根本没有学习气氛，优哉游哉每周开会讨论文档，开始写代码后才发现时间有多急，最后一些功能可能处理得也不是很好。目前来看我们的实现还比较简陋，而且查询效率比较低，也是第一次自己架构数据库考虑不够全面，后来顺便上刘嘉老师的数据库开发课才想到有挺多能优化的地方，而且还应该为迭代二迭代三考虑考虑。后面的任务要早些开始了，要努力比迭代一完成的好，加油加油。

刘焱 171250534

迭代一总的来说内容不多，要写的文档索然有好几个，但是有了软工二的经验后写起来还是挺简单的，而迭代一的功能点也不算多。但是，真正从零开始实现起来，还是让人感到头大，虽然可以借鉴软工二的大体框架，但里面的实现却是真的需要自己来填。

我的分工是和另一位同学一起负责后端，但我们的配合还做的不太好，有点眼神队的味道。对我来说后端问题最大的就是数据库操作部分的xml了，因为以前一直没学习过xml，而且数据库的知识忘得太快了，导致老是写不好，而且我们的数据库构建没能一次快速地完成，导致后续的sql代码又需要更改，很麻烦。

总的来说，与团队一起，从零开始编写一个项目，虽然还只是迭代的第一阶段，但其中的困难还是让我有不少的收获。对于Git的使用，对于小组遇到的困难的讨论，对于遇到工具版本不同造成的问题的解决，这些都是一个人编程单打独斗所不能得到的。

有了迭代一的经验，我相信在接下来的迭代中我们应该能够实现得更加流畅，更加有默契，能够将这个项目做到更好，做出我们的特色。

许月洋 171250571

作业还是早写的好，虽然慨叹无数遍，一直有拖延着，作为本组划水选手，只完成了数据的初始化，仍有些磨磨蹭蹭，仍需努力。

初步做完数据库导入的工作后，bug很多，因为不严谨导致的，本来想通过后端直接把数据导入，绕开前端的。具体尝试是自己声明一个初始化对象，用这个对象去初始化数据，这就导致前面一半没有用依赖注入，后面一半试图用mapper等之类，程序怎么跑也跑不起来，也不太懂原理，调试了很长时间，最后和组员商量通过前端一个按钮来实现初始化，方便了很多。

在另一门服务端的课上，老师展示了封装的jdbc模板插入数据是多么的方便。在作业的一开始我便去尝试，甚至试图将老师的配置全盘搬过来，可是又不太懂xml的各种配置，照搬过来后仍然无果，便去网上看各种jdbc的教程，配置，越看越乱，花了很长时间写的有关jdbc代码因为配置错误问题，错误不断，最终放弃。果断采用软工2使用的mybatis。当写完之后发现，，jdbctemplate在大规模写入数据的时候并不一定比mybatis方便多少，甚至jdbc模板的代码还更繁琐一点。感想就是在不熟悉各种配置信息时，不要去尝试

陌生的东西，还是用熟悉的框架去完成作业，不然只会浪费很多时间。等多了解之后再去用新的。

关于数据库的感想，就是建立数据库的感想，就是数据很多，很乱。在一开始设计表格的时候，由于不太懂学术网站的各种信息的意思，相当于跨行工作，得先了解各个术语的意思。因此建立数据库较为麻烦，这就让我想起了面向服务的软件工程中讲的内容，真的太有用了。所以让IT专家和行业专家结合开发的思想很重要。写入数据的时候由于项目众多，而且excel表格看的眼花缭乱，甚至无中生有了一列出来了，所以眼杂的工作还是要队友帮忙检验为好。数据库一开始的设计对以后影响重大，我认为这次我的设计就有些冗余了，导致运行有一些缓慢。而且一旦哪儿一开始写错了，更改的过程特别麻烦，很多名词名称相近，盯着电脑看久了很崩溃，眼睛看花了更容易出错，所以要多动一动，保持状态，少出错。前期少出错，后期就会省去更多的精力去修改。