项目二实战总结

* **项目启动：**

周三开始项目，周五晚上结束战斗。本来觉得这个项目三天完成有点难，上手以后觉得还可以，主要大部分工作组长都给做了，架子搭建好。只要写业务逻辑就行。本次项目我负责的是设备模块。一共5个小模块，首先拿到项目，将数据库进行导入，然后对自己对应模块的数据库进行的分析如下：

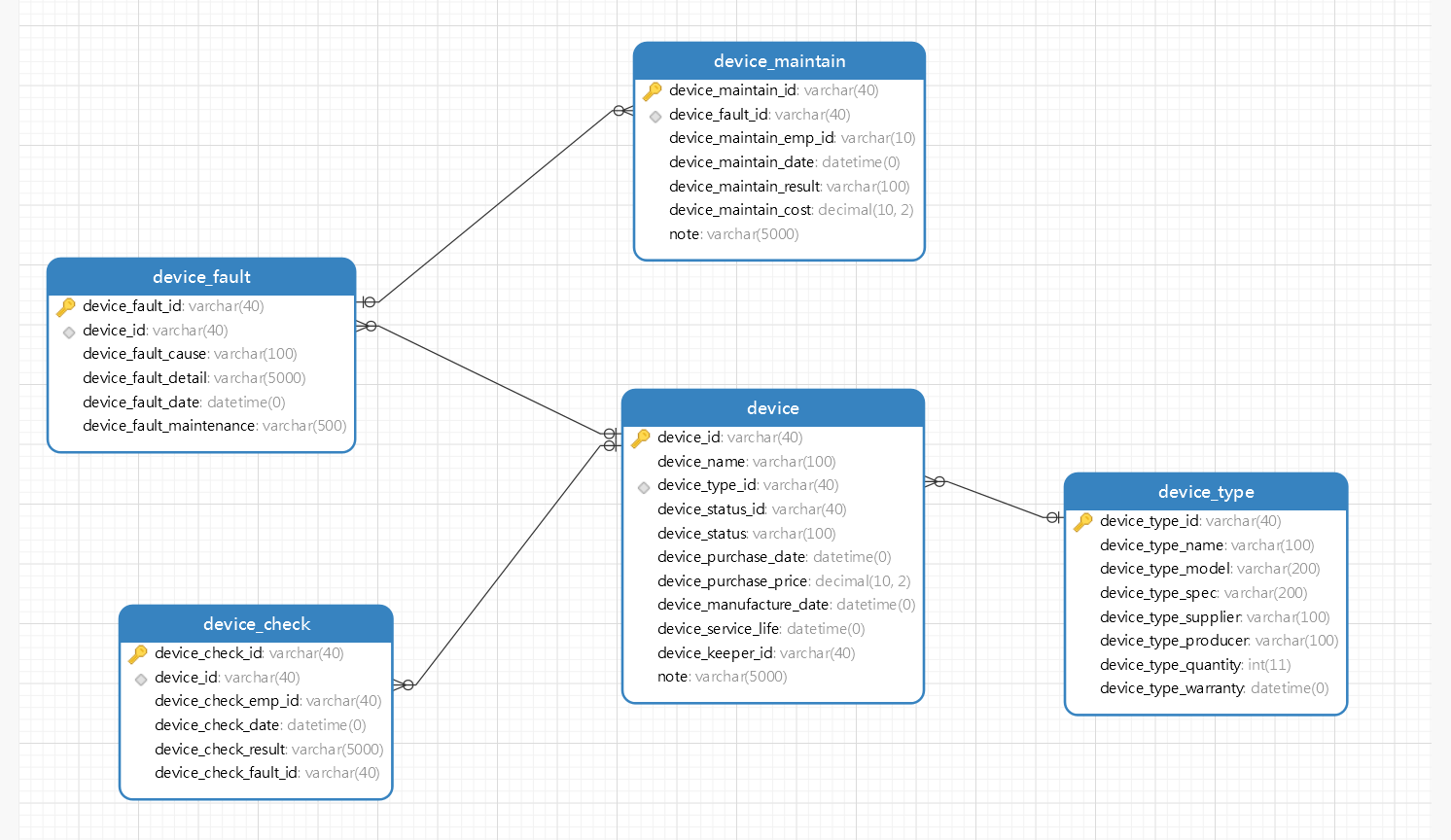


图1

对自己模块的每张表进行分析，分析发现。最右边的设备种类是个单表，不涉及其他表。首先需要写的东西。然后是中间的设备表，很重要的一个表，基本上涉及到其他每一张表。这个表的外键需要的是设备种类的表。然后其他三张表需要设备表。所以设备表的模块第二个完成，之后再就是检查模块，需要到设备表。之后设备故障模块需要设备表。最后是设备维修模块，需要到设备故障表。所以最后确定下来写模块的顺序是：设备种类、设备、设备检查、设备故障、设备维修。

等待组长搭好架子，clone然后从前端代码入手。第一眼看到这个前端的时候，what？这是什么鬼？完全是一个新的前端，网上搜了一下，似乎是用的EasyUi写的前端。首先需要知道是这种写法的前端是怎么进行传数据和接收数据的。最先想到的是用搜索引擎，找到EasyUi的官网，简单看了一下，找到的都是这个前端怎么进行使用，各种组件需要用到什么样的class，并没有找到怎么进行数据的传输的。随后老师反复强调通过抓包来获取如何通过数据的传输。之后就进行抓包查看，因为有模板，只需要查看模板里面的数据是如何传输的，然后模仿就行了。之后的工作就是使用ssm进行功能的实现了，复习了前面ssm的内容。Spring、SpringMVC、Mybatis整合在一起时是如果使用。相应的文件的配置，注解的使用以及插件的使用等等。

* **Bug汇总及解决：**

1. 日志是个好帮手，能够帮我们搞清楚框架这个黑匣子的运行状况，也可以避免使用System.out输出到控制台，造成其他模块控制台出现没必要的信息
2. Mybatis的动态Sql能够使代码更简洁，接口不用写太多，比如模糊搜索时，使用REST风格的url接收不同的条件，使用动态SQL的判定条件进行相应的查询，只需要写一个controller方法和一个标签就可以查各种条件
3. 使用AOP，将无关具体业务的代码提取出去，比如将需要分页的功能使用AOP统一进行处理
4. Spring的xml配置有时候会比javaConfig更清晰，更好维护；javaConfig使用方便，但团队最好统一意见，否则会遇到冲突的情况
5. Sql中的表的列名与Model的字段、Json数据与Model的字段对应时，类型和名字没有对应造成的错误都不易察觉，往往造成大量时间的浪费，应当尽量避免手写，遇到奇怪的错误时也需要关注这一块
6. 整个业务逻辑是一个数据的递交换，中间的媒介是model层，其中使用反射调用set、get方法很频繁，所以要注意不能缺失

* **收获及总结：**

总的来说这个项目没有第一个项目的收获大，因为是小组合作完成，很多的事组长都给你安排好了，然后给了模板，什么东西不会，抓包看比较数据哪里不对，然后做相应的改正就好，遇见bug抓包看log。所以值得学习的是这个项目是怎么样实现的。想想如果没有模板，不给现成的项目给你参考，以及没有前端的url提示，我是否还能够很清楚的理清每个功能项目是怎么巧妙的实现的。用什么接口，方法怎么规范的命名，项目的整体架构是怎样的，在脑海中有个清晰的认识。其次是git的使用，在遇见冲突的时候怎么解决相应的冲突。明白在整个项目中自己的位置，自己需要别人什么样的接口，以及为别人提供什么样的接口等等。