

# 开发体会 & 小结

姓名：沈韵枫

专业：软件工程

学号：3200104392

## Pre

在本学期的学习过程中，我使用 `Flask` 框架完成了“智能家居管理系统”的搭建，支持基本的用户注册与登录验证功能，同时也支持用户在界面上对部分元素进行拖动、更改，以实现智能家居设备的摆放与状态编辑。此外通过可视化方式展示了设备列表信息，较为清晰直观。

虽然功能不甚丰富，但也勉强符合预期。架构经历了几次较大的改动，也让我清楚的认识到了实际开发过程中的“计划赶不上变化”吧。

## 后端开发

### 多级路由

在使用 `Flask` 框架进行开发的过程中最早打算使用一级路由（见设计报告）。由于存在较多同质操作（增添/删除），为了对不同的行为对象进行区分，`URL` 就会变得很长，同时 `app.py` 文件也会变得极长、不方便阅读。经过查阅资料后使用了 `Flask` 提供的 `blueprint` 将不同功能的代码分散到不同文件中，最后注册到应用程序中实现汇总。同时产生了二级路由，功能与路径的对应较为直观。

### 数据库操作

在对数据库的操作上，最初是使用原生 `SQL` 语句进行实现的，导致了大量语义重复的代码，且在拼接字符串的过程中及其容易出现缺少单引号，缺少参数之类的细节错误，为开发过程带来了诸多不便。在查询相关资料后改为使用 `Flask-SQLAlchemy` 进行实现，使得实现过程可以更加聚焦于业务逻辑，而非 `SQL` 语法。同时，`Model` 的存在也使得对于记录的操作更加便捷、更语义化，让数据库的操作不再繁琐（虽然刚开始配环境废了好一阵功夫）。

### 接口测试

尝试使用 `Postman` 作为测试工具，以实现前后端开发的解耦（虽然这一设想由于整体架构的变更最后还是被破坏了）。分组储存测试样例的功能非常的便捷。参数、请求方式的修改也非常的高效，但是发 `json` 需要手动在 `header` 中添加说明还是比较痛苦的。在从采用直接通过 `POST` 请求 `body` 部分传递参数，向使用 `ajax` 发送 `json` 包进行请求的转变过程中卡了好一会儿。（但好在转换方式后，后端解析 `request` 的效率得到了提升）。

## 跨域

由于是前后端不分离项目，此处直接在后端使用 `CORS` 库进行了粗糙的解决——简单粗暴的允许来自所有 origin，采用所有请求方法的跨域请求。

## 前后端对接

在前后端交互部分，本来计划在服务端维护一个“已登录用户列表”，在前端每次发送请求时捎带用户id进行验证，但因为不会在前端存储数据而不了了之。在查询相关资料后选择使用 `session` 进行实现，同时补充实现了对未登录用户的拦截与重定向功能，增强了平台的数据安全性。

在接口设计上，由于后端的开发工作与设计文档的撰写工作是在前端实现之前进行的，导致大多数数据的返回格式与前端预期的处理格式不一致，导致了较长时间的返工。这一点会在以后的开发过程中注意。

此外，由于前后端开发工作之间的间隔较长，而在后端开发过程中没有注意撰写相应的接口文档（因为实际实现与返回格式较设计文档中略有不同），导致进行前端开发时需要反复查看后端代码，降低了工作效率。在之后的开发工作中，也应当注意过程文档的撰写，以提高交互过程的效率。

## 前端开发

---

### 几次改动

最初计划采用 `Vue + Flask` 实现一个前后端分离项目，网络请求方面使用 `axios` 库进行实现，但最后由于没有充足的事件对前端框架的使用进行学习，无奈采用了前后端不分离的开发模式，请求方面也换用了 `jQuery` 封装的 `ajax`，导致了前后期撰写的 `js` 文件代码风格较为混乱——`jQuery` 语法和原生语法混杂，可读性堪忧，但好在还是保障了基本功能的实现。

在前端页面布局方面，最初在 `Figma`（[智能家居管理系统页面原型](#)）上进行了整个平台的原型设计，计划采用下拉列表、多级菜单等方式提供更加简洁直观的交互方式，使得用户在不需阅读《使用手册》的情况下也能掌握平台的使用方法。但由于本人的 CSS 水平低下，并且在页面布局的设计上出现了失误，导致无法按预期方案实现“场景选择”功能设计的下拉菜单，以及左侧设备列表中的“收起/展开”功能，略显遗憾。

### “画板”功能的实现

由于没有使用框架和组件库，这部分完全是手工通过原生 js 实现的，通过监听 `mousedown`、`mousemove` 和 `mouseup` 计算每次鼠标运动的偏移量，并对相应的图标位置进行修改，同时以画板位置对偏移量的最大最小值作出限制，防止用户将图标拖到预期外的位置。图标的保存与恢复则通过与画板的相对定位实现。

显然，该实现方法不甚优雅，对于不同尺寸设备的适配也不太能行。此外也没有做节流和防抖的工作，导致实现效率低下，希望之后可以找到更好的实现方式吧。

## Summary

---

在本次实验过程中，我首次独立完成了一个 Web 产品的前后端开发工作，从只能开发用于展示的“静态”界面走上了可以进行交互的“动态”界面上。虽然在 UI、设备适配等方面仍存在诸多不足，整个开发过程也令我获益良多，加深了对整个 B/S 架构的了解。

此外，“智能家居管理系统”是一个具有极大的实际意义的选题，希望将来能真正实现和联网设备的联动，成为一个真正可用的 Web 平台。在寒假也会尝试将整个项目迭代为前后端分离项目，同时适当使用一些组件，让项目代码的可读性、健壮性得到提升。