

# 系统实现报告

---

## 一、实现环境

---

### 1.前端技术栈

- Vue————3.2.36
- axios————1.2.0
- Vuex————4.0.2
- Vue-Router————4.1.6
- Naive-UI
- Element-UI
- Echarts

### 2.后端技术栈

- MySQL

### 3.Github

[alonelysnake/BUAA-Database-Vue-Frontend: 数据库大作业Vue前端 \(github.com\)](https://github.com/alonelysnake/BUAA-Database-Vue-Frontend)

## 二、系统功能结构图

---

- 用户功能模块
- 管理员功能模块

## 三、数据库基本表定义及约束

---

基本表的定义，主外码等完整性约束定义，索引的定义

## 四、系统安全性设计

---

## 五、SQL代码说明

---

### 1.存储过程

### 2.触发器

### 3.函数

## 六、主要技术和模块

---

## 1.后端映射数据库实体

## 2.后端设计模式

## 3.前端单页面路由

为实现单页面应用，使用Vue-Router来管理路由以及该路由下挂载的组件。在router/index.js下，设置了所有路由和组件的对应关系。路由发生切换时，只销毁并挂载发生变化的组件，未变化的组件保持不变，最大化的减少了前端重复挂载的开销以及由此产生的对后端数据库的不必要请求。

## 4.前端组件动态渲染

用户对页面的操作和前后端异步执行时导致的数据动态变化需要保证组件能随数据的变化而呈现不同的展示。本项目通过ref()函数实现了组件与数据的动态绑定，从而根据数据变化渲染出不同的结果。所有变量都用ref包裹成一个对象后绑定到html组件上，通过修改其中的value属性改变具体值。不论是用户在页面中的操作还是代码执行阶段的赋值，都是在对同一片空间进行修改，由此实现了前端的数据协同。

## 5.前后端数据通信

前端部分中，在utils/request.js中定义并暴露了一个对象request。该对象拥有post和get方法，用于向后端发送和接收数据。进行通信时，前端首先设置好两个参数：要调用函数的路由、按照json格式的字符串化的数据。然后调用request对象的方法向后端发出请求。考虑到后端的处理延迟，我们采用异步的方法进行通信。后端处理完成返回结果时，在then()函数中用箭头函数来接收必须同步执行的指令。对于其他不必与这个请求同步的指令，前端发出请求后可以继续顺序执行，不必等待。

后端会以路由的形式暴露出一系列的接口函数。前端调用的接口函数会尝试接收并解析前端发送的参数，进行处理后再按照json格式返回。因为所有参数都是键值对的形式，所以只要前后端设置了相同的key，就可以得到期望数据。示例如下：

```
let post = JSON.stringify({
  'username': name.value,
  'email': mail.value,
  'password': md5(password.value),
});
request.post("register/", post).then(res => {
  //后端的返回结果
  let recvMessage=res.message;//接受后端返回信息
  let success = recvMessage==='注册成功';
  if (success) {
    message.success('注册成功');
  } else {
    message.error(recvMessage);
  }
});
```

对应的后端部分代码：

```
def register(request):
    content = request.body.decode()
    content_dict = json.loads(content)
    name = content_dict.get('username')
    email = content_dict.get('email')
    password = content_dict.get('password')
    ...
    data = {'message':message}
    result = JsonResponse(dict(data))
    return result
```

## 七、系统功能演示

## 八、源程序简要说明

## 九、亮点介绍

## 十、收获与体会

## 十一、小组成员分工

	王永瑶	李京昊	卓乐
系统设计阶段	1. 前期调研、分析系统需求设计 2. 数据库系统的概念模式（E-R图） 3. 设计数据库关系模式 4.根据需求设计前端网页基本布局与模块	1. 前期调研、分析系统需求设计 2. 数据库系统的概念模式（E-R图） 3. 编写数据流图与数据元素表 4. 根据需求设计前端网页基本布局与模块	1. 前期调研、分析系统需求设计 2. 数据库系统的概念模式（E-R图） 3.设计数据库系统的逻辑模式并将所有关系规范化到 3NF 4.设计后端架构与数据库的主要功能
系统实现阶段			
系统报告撰写			