

课程实验报告

课程名称：软件架构与应用开发

课程选题：小微企业跨组织人才管理系统

小组成员： 王灵霜、柴巧霞

学 号： 31801061

姓 名： 王灵霜

专业班级： 计算机1801

所在学院： 计算机与计算科学

报告日期： 2021 年 6 月 29 日

一、作品设计

目标：建立一个小微企业跨组织人才管理系统，hr可以建立员工档案，评价员工，查看员工及其工作过的公司对其的评价，来进行人才管理和招聘。

功能：

1. 区分hr和管理员注册登录
2. hr能对员工进行增改查、评价和招聘解雇功能
3. 管理员能对公司和员工进行删改查功能
4. 系统结构描述

cache

-- RedisCache 自定义cache接口实现rediscache

controller 接收请求调用service

-- AdminController /admin

/register post 管理员注册

/login post 管理员登录

/updatePwd get 修改密码

/select get 查看管理员

-- CompController /company

/addUnemployed get 添加无公司

/register post 注册公司

/login post 公司登录

/select get 查看公司

/update post 修改公司

/updatePwd get 修改密码

/findAll get 获取所有公司

/delete get 删除公司

-- EmplController /employee

/add post 添加员工

/select get 查看员工

/update post 修改员工

/findByCompId get 获取本公司员工

/findOther get 获取非本公司员工

/recruitEmployee get 招聘员工

/dismissEmployee get 辞退员工

/delete get 删除员工

/findAll get 获取所有员工

-- EvalController /evaluate

/add post 添加评论

/findByEmplId get 获取简历评论

/deleteEvalById get 删除单条评论

dao 数据库操作

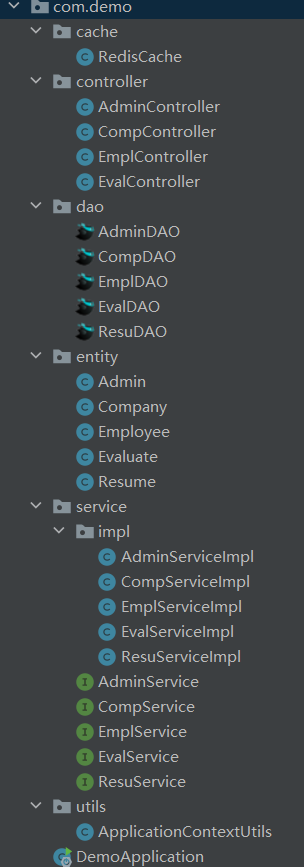
entity 实体

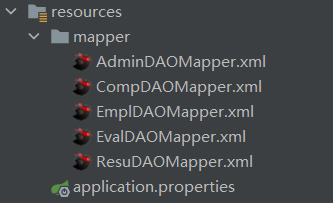
service 封装业务便于controller调用

utils 工具类

-- ApplicationContextUtils 获取springboot创建的工厂

后端项目结构

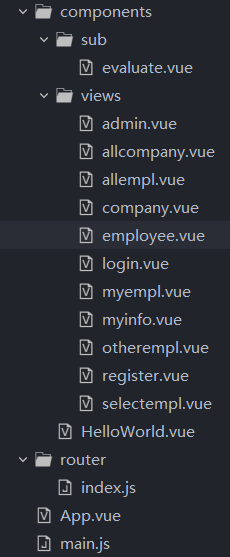




DAO的Mapper对象

配置文件

前端项目结构



subs 编写的组件

-- evaluate 评论列表项

views 页面

-- admin 管理员主界面

-- allcompany 公司管理

-- allempl 员工管理

-- company 公司主界面

-- employee 员工信息

-- login 登录

-- myempl 我的员工

-- myinfo 我的信息

-- otherempl 招聘员工

-- register 注册

-- selectempl 公司员工

1. 实现重点

本次项目中我主要负责数据库的编写，前端的编写，后端接口的编写和改进

下划线为我的编写，加粗为我的改进

-- AdminController /admin

/register post 管理员注册

**/login post 管理员登录**

/updatePwd get 修改密码

/select get 查看管理员

-- CompController /company

/addUnemployed get 添加无公司

/register post 注册公司

**/login post 公司登录**

/select get 查看公司

/update post 修改公司

/updatePwd get 修改密码

/findAll get 获取所有公司

/delete get 删除公司

-- EmplController /employee

**/add post 添加员工**

/select get 查看员工

**/update post 修改员工**

/findByCompId get 获取本公司员工

/findOther get 获取非本公司员工

/recruitEmployee get 招聘员工

/dismissEmployee get 辞退员工

/delete get 删除员工

/findAll get 获取所有员工

-- EvalController /evaluate

/add post 添加评论

/findByEmplId get 获取简历评论

**/deleteEvalById get 删除单条评论**

redis的实现

工具类ApplicationContextUtils

@Component

public class ApplicationContextUtils implements ApplicationContextAware {

private static ApplicationContext applicationContext;

@Override

public void setApplicationContext(ApplicationContext applicationContext) throws BeansException {

this.applicationContext = applicationContext;

}

public static Object getBean(String name){

return applicationContext.getBean(name);

}

}

RedisCache

@Slf4j

public class RedisCache implements Cache {

private String id;

public RedisCache(String id){

log.info("当前缓存的id:[{}]",id);

this.id = id;

}

@Override

public String getId() {

return this.id;

}

//放入redis缓存

@Override

public void putObject(Object key, Object value) {

log.info("放入缓存的key:[{}] 放入缓存的value:[{}]", key, value);

getRedisTemplate().opsForHash().put(id,key.toString(),value);

}

//从redis缓存获取

@Override

public Object getObject(Object key) {

log.info("获取缓存的key:[{}]", key.toString());

return getRedisTemplate().opsForHash().get(id, key.toString());

}

//删除指定的缓存

@Override

public Object removeObject(Object o) {

return null;

}

//清除缓存

@Override

public void clear() {

log.info("清除所有缓存");

getRedisTemplate().delete(id);

}

@Override

public int getSize() {

return getRedisTemplate().opsForHash().size(id).intValue();

}

public RedisTemplate getRedisTemplate(){

RedisTemplate redisTemplate =

(RedisTemplate) ApplicationContextUtils.getBean("redisTemplate");

redisTemplate.setKeySerializer(new StringRedisSerializer());

redisTemplate.setHashKeySerializer(new StringRedisSerializer());

return redisTemplate;

}

}

前端使用webpack+vue2+elementUI编写，使用axios异步调用接口

示例

this.axios.get("http://localhost:8888/A-Demo/employee/findByCompId?id="+this.$route.params.id).then(

res => {

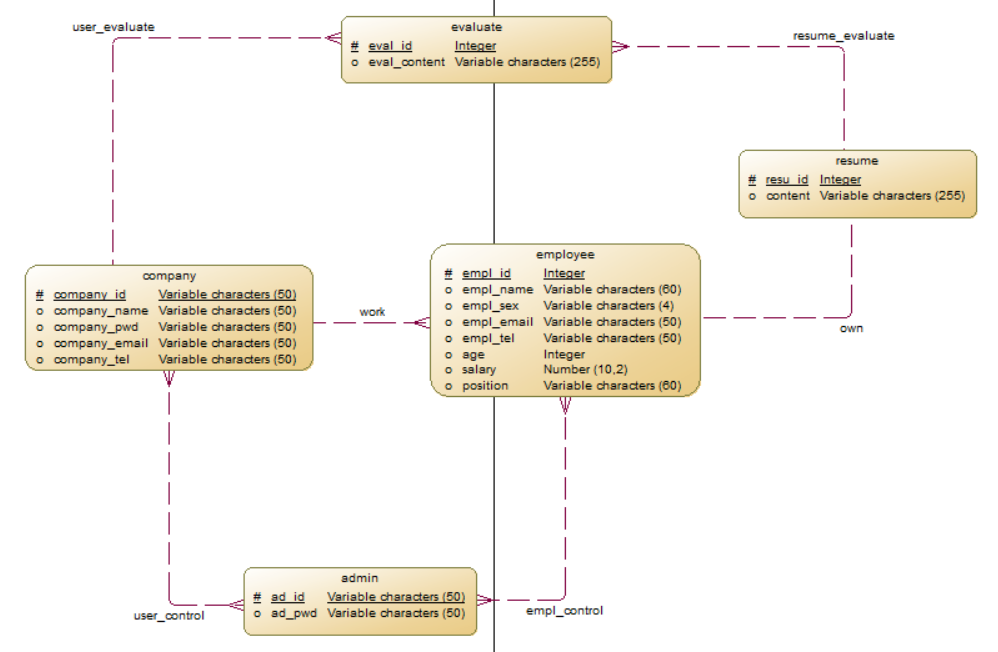
this.EmplData = res.data;

})

数据库模型如下

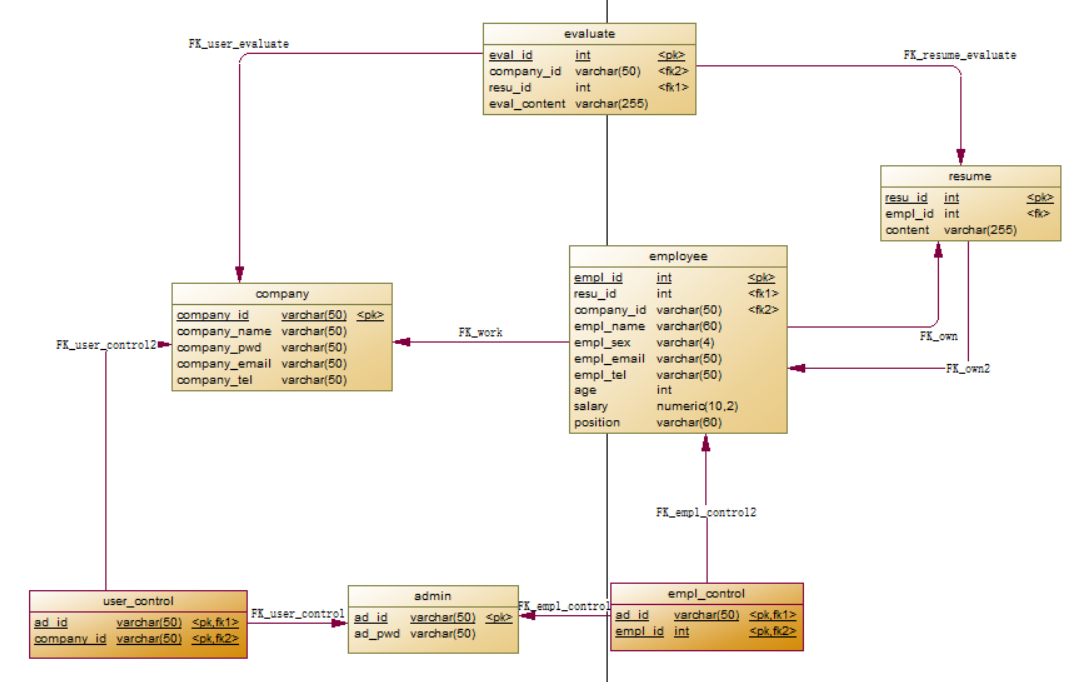
概念模型设计

包括公司、员工、评价、简历和管理员共五个实体，其中公司发布评论为一对多关系，公司雇佣员工为一对多关系，员工和简历为一对一关系，简历和评价为一对多关系，管理员管理公司和员工为多对多关系。

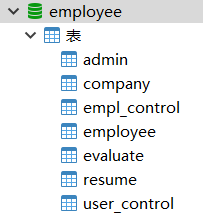


物理模型设计

由概念模型直接生成，并在评价、员工和简历表设置了主键自增



生成的数据库表



1. 运行结果
2. 注册页面

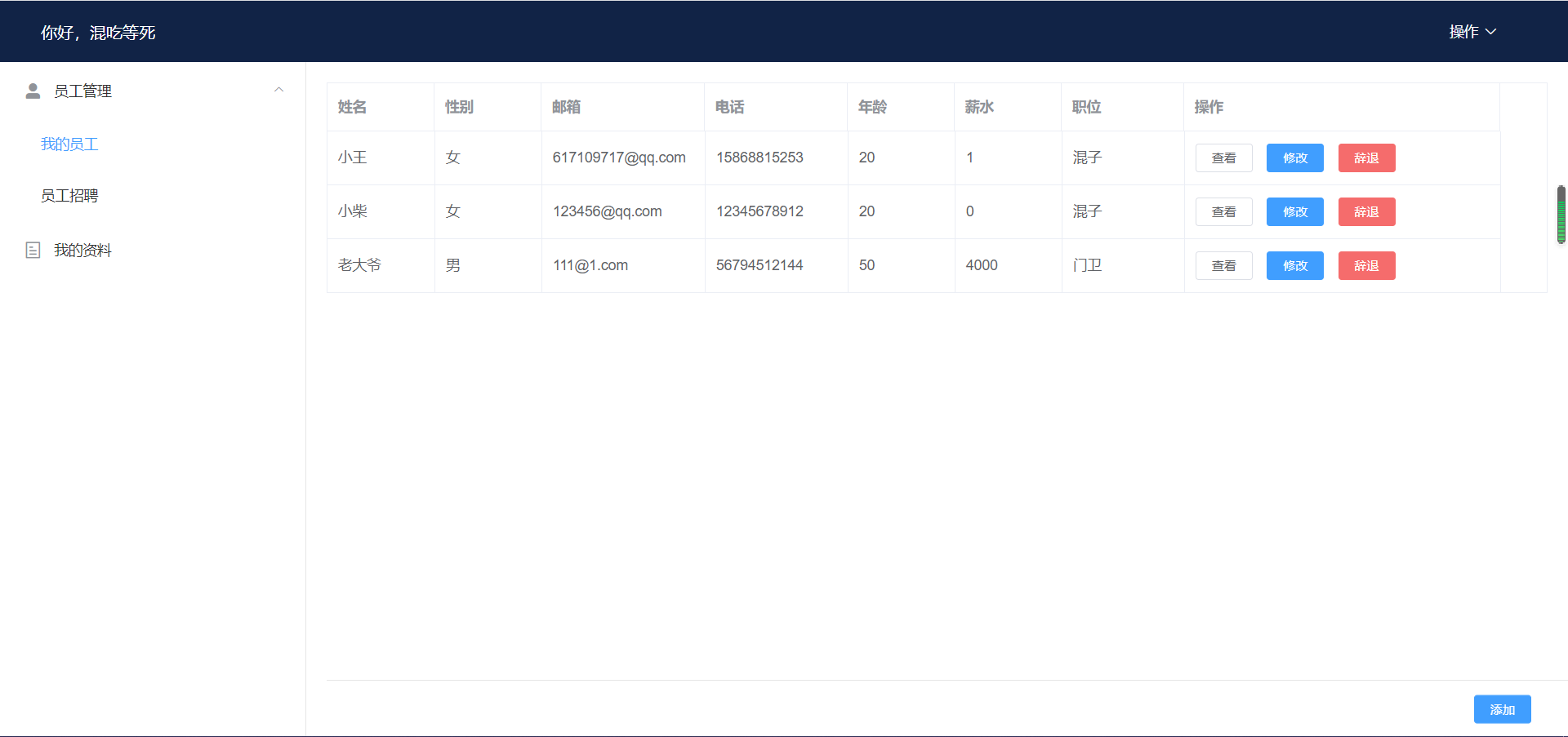


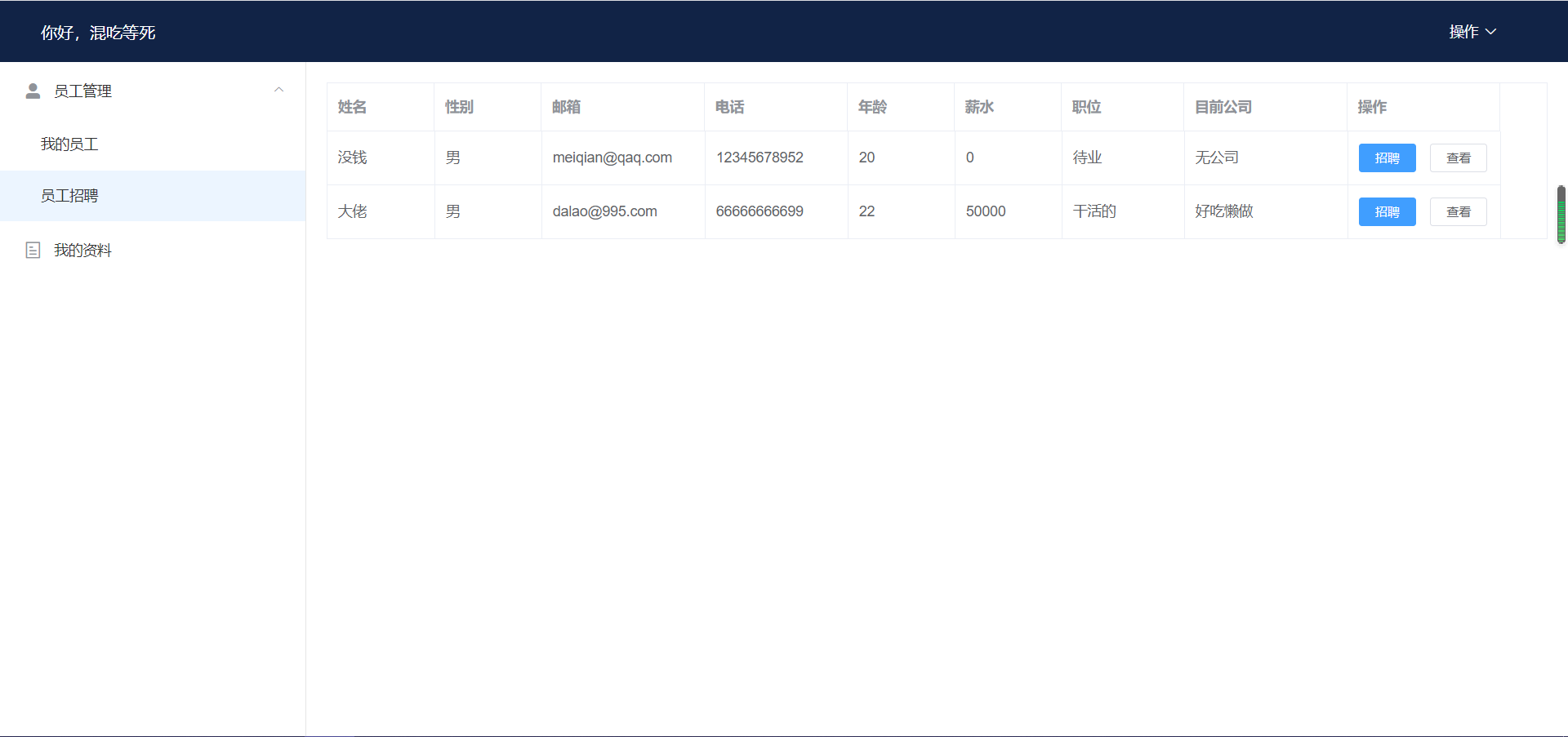


1. 登录页面



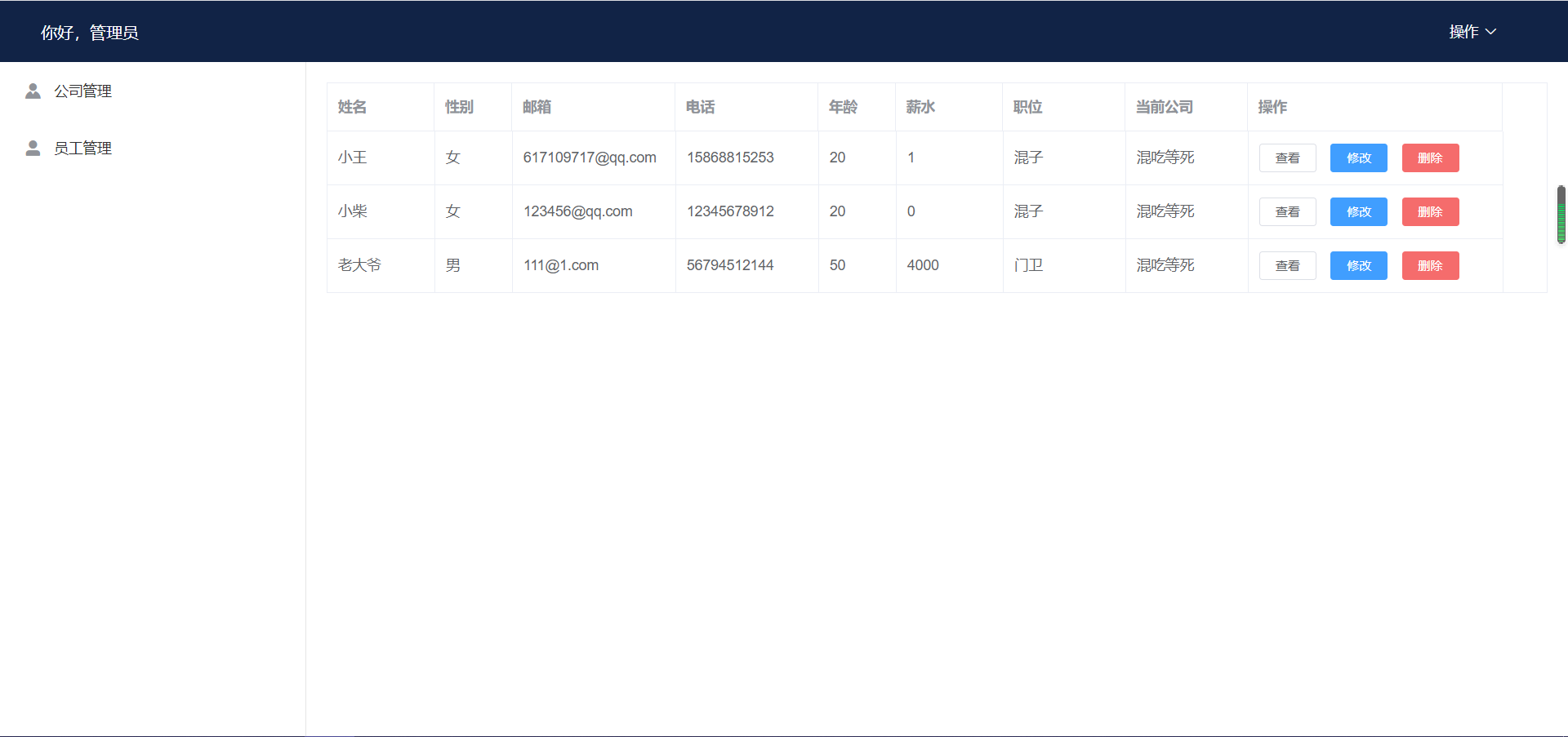
1. 公司界面



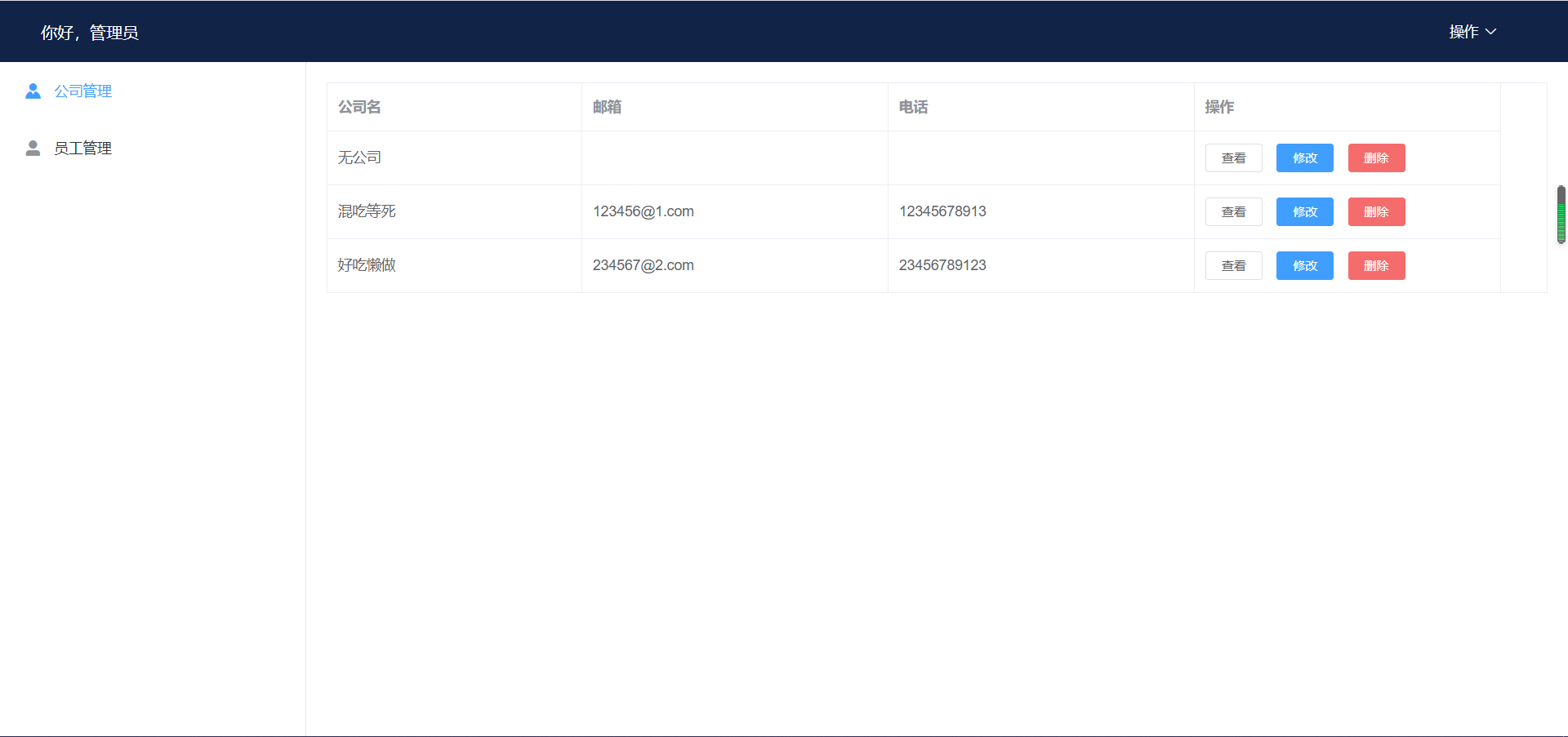


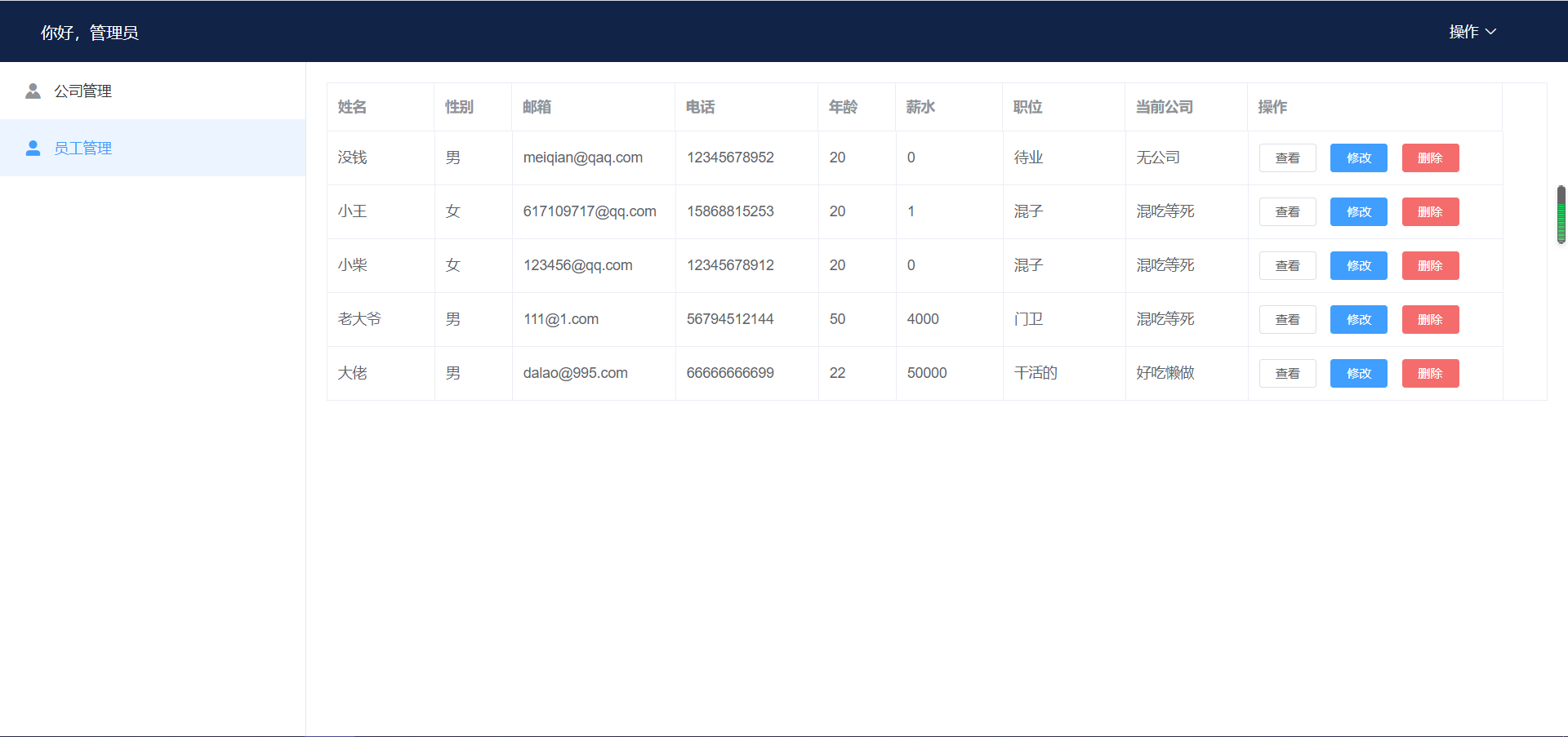






1. 管理员界面







1. 总结

很久不用powerdesigner画模型建数据库了，都忘得差不多了，通过这次的项目又重新熟悉起来，项目做到现在功能算比较完善了吧。因为自己写前端，十分需要错误提示，不能数据没调到后台报错前端没返回，所以重改了很多接口的返回。

遇到的一些困难：

1. 一开始直接概念模型一路生成数据库，没有主键自增，后面查询资料了解了需要在物理模型中单独修改属性。
2. 员工简历和员工信息一对一的关系，互为外键，所以增删员工操作都非常麻烦，增加员工需要先建立员工，再建立简历，再修改员工表中的简历id，员工的删除需要先把该简历的评论全部删除，然后把员工的简历id置空，再删除简历，再删除员工，艰难的把步骤写死了因为不想再改数据库
3. redis一直报错说无法创建bean，后面证实是配置文件和依赖的问题
4. redis本地启动连接过多无法启动，命令行清空，重新建立连接