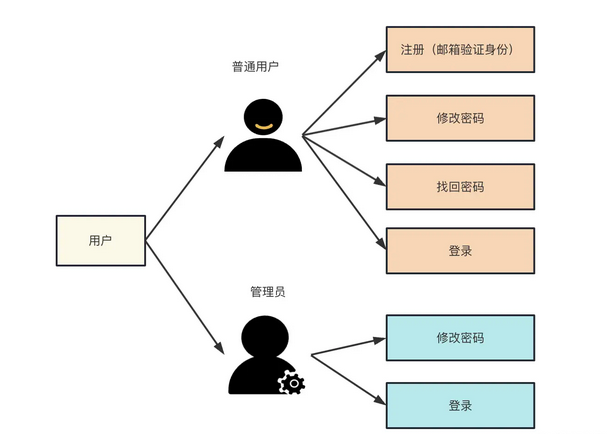
**一、会议室预约系统需求分析**

这个项目有两个端、用户端和管理后台。

用户端(前端)：有两种用户角色：普通用户和管理员、各自有不同的功能。

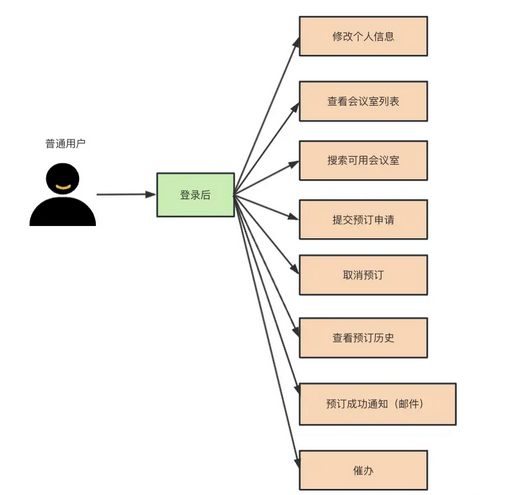
普通用户：可以注册，注册的时候会发邮件来验证身份，注册之后就可以登录系统了。

管理员：账号密码内置，不需要注册。

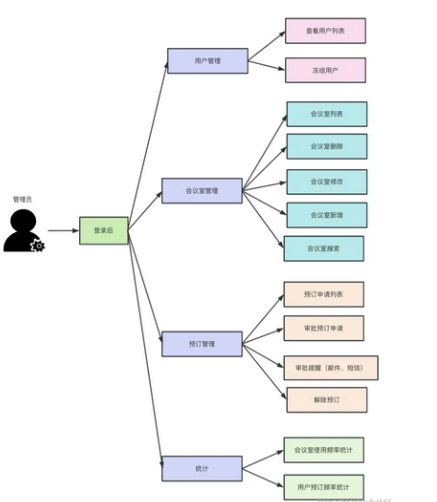


**功能：**

普通用户可以查看会议室列表、搜索可用会议室、提交预订申请、取消预订、查看预订历史等。预订成功会邮件通知，就是注册时的那个邮箱。如果管理员一直没审批，可以催办。



管理员：可以查看用户列表、冻结用户、会议室列表、搜索会议室、添加/修改/删除会议室、审批预订申请、查看会议室统计信息等。

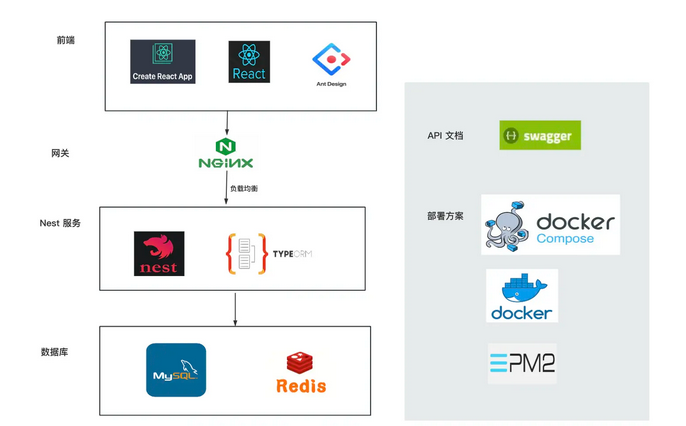


整个系统整体分为用户管理、会议室管理、预订管理、统计这 4 部分。如果超过 10 分钟没审批，会发送邮件提醒管理员，如果超过半个小时没审批，会发送短信。管理员可以解除用户的预订，释放会议室。冻结用户是指把用户设置为冻结状态，冻结状态的用户不能预订会议室。统计模块会按照会议室维度和用户维度进行统计，并报表展示。

具体原型有空再自己出一版。

**二、技术方案**

前端做两版：一种是 React + AntD、一种是Vue3 + ElementuiPlus。后端统一使用Nest.js。



前端使用 create-react-app 脚手架创建项目，使用 react + antd 开发页面。

使用 nginx 来网关层，实现静态资源的托管，并且对于动态资源请求做负载均衡。

使用 Nest 开发后端业务逻辑，使用 TypeORM + mysql 来做 CRUD。

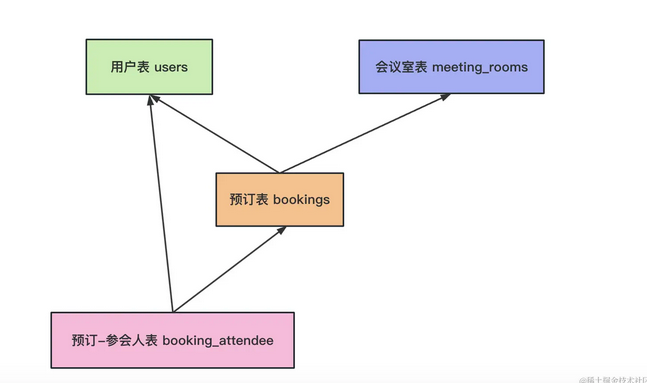
使用 Redis 来做缓存，减轻数据库的压力，提高响应性能。

api 文档使用 Swagger 来生成。

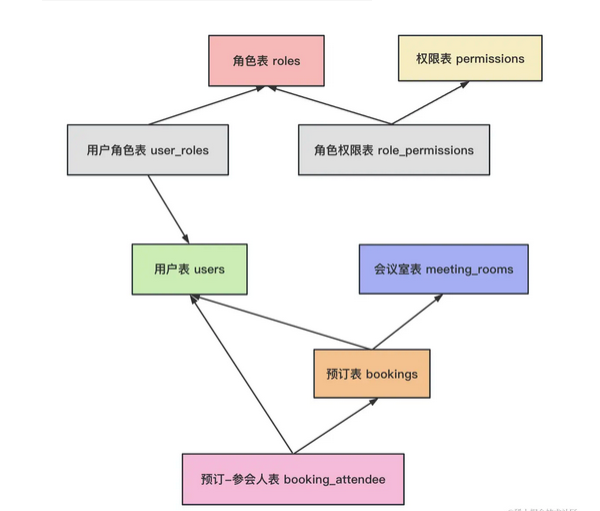
部署使用 Docker Compose 的方式。

PM2 可用可不用，不用 PM2 就要启动容器的时候指定重启策略了，这里为了简单我们还是用 PM2。

1. **数据库设计**



再加上角色表、权限表，还有 2 个中间表，完整关系如下：



1. **模块API设计**

**用户模块**



**会议室管理模块**



**预订管理模块**



**统计模块**



1. **角色划分**

权限控制使用 RBAC 的方式，有普通用户和管理员两个角色