\mathcal{T}

智能会议室管理系统

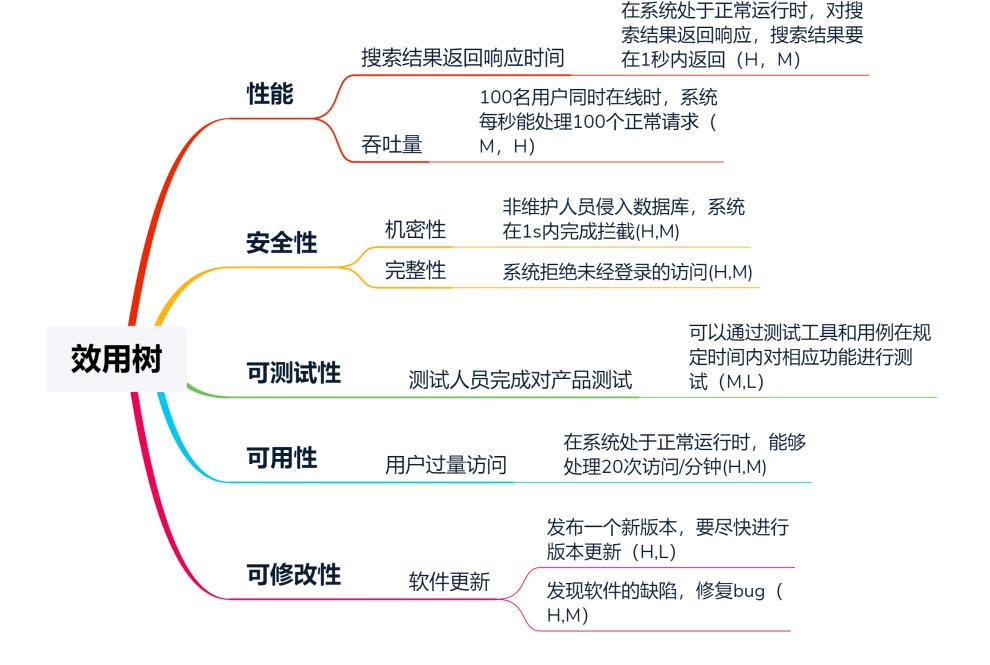
付彤,白振宇,刘孟骁,王宇辰,周韬

Utility Tree

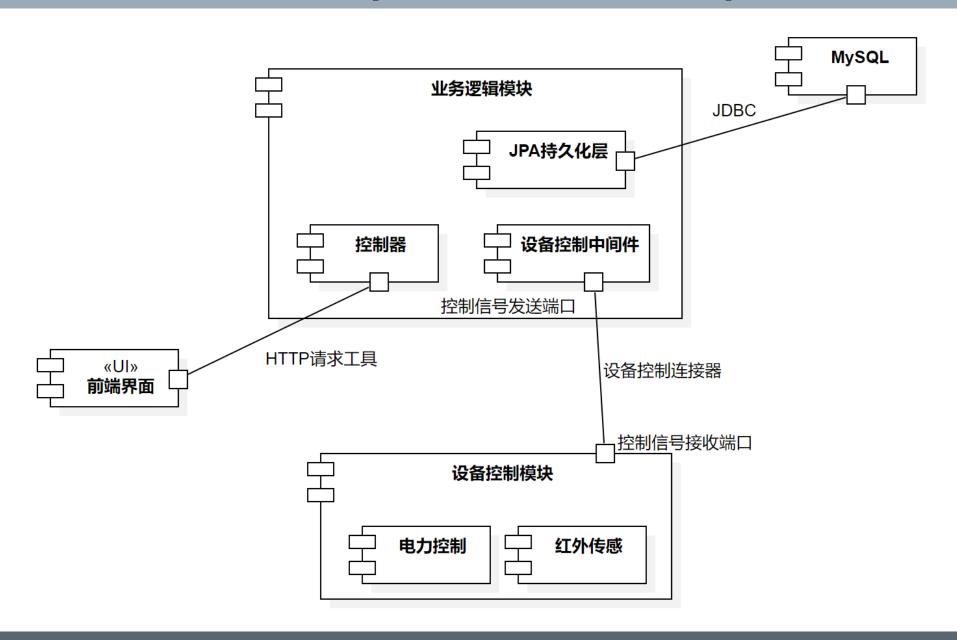


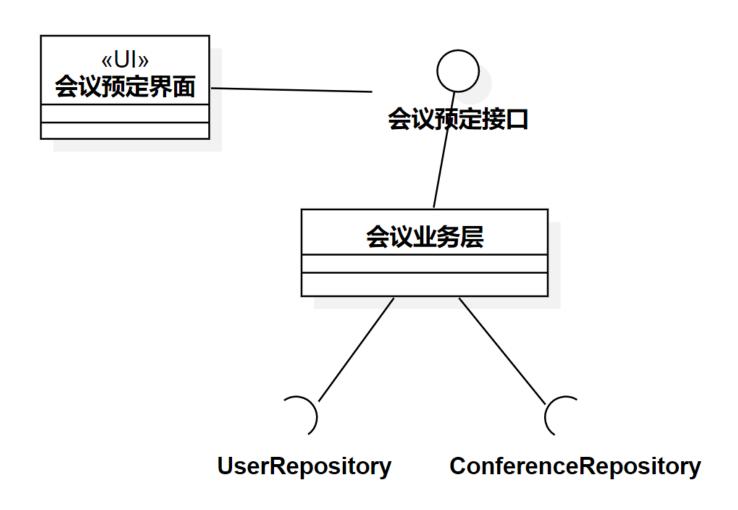
软件体系结构描述

软件体系结构风格

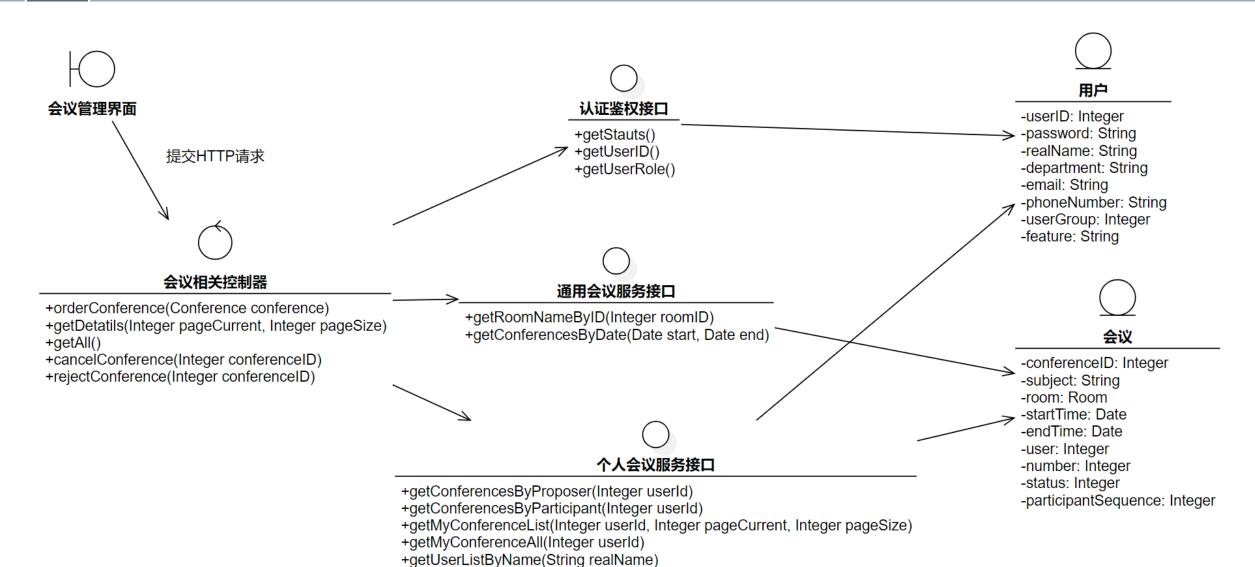


组件和连接器 视图 (以业务逻辑层为例)

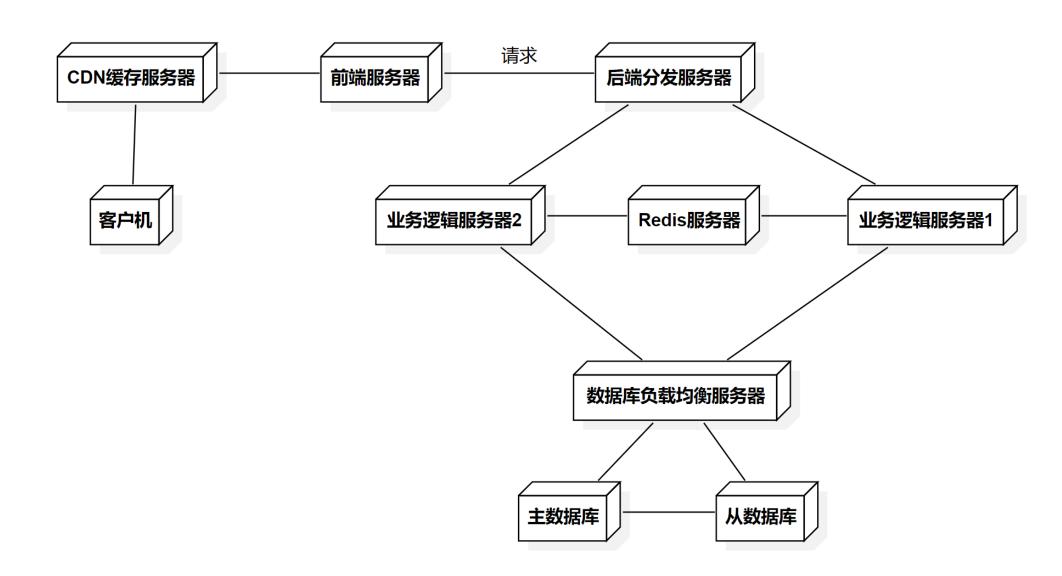




模型视图 (以会议管理设计类模型为例)



+orderConference(Conference conference)

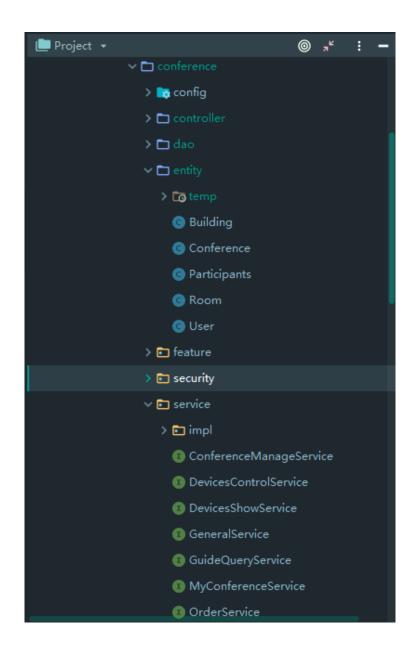


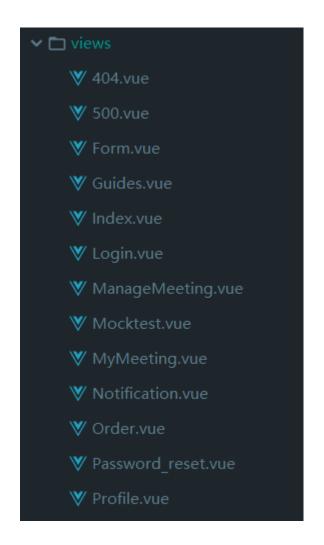
- ➤ MVC分为模型(M),控制器(C)和视图(V),以插件式、松耦合的方式连接起来。
- ▶ 持久化模型负责业务对象与数据库的映射(ORM), Web视图负责展示信息和同用 户交互,业务控制器接受视图传来的用户的输入调用并模型完成特定功能。

在我们的项目中,

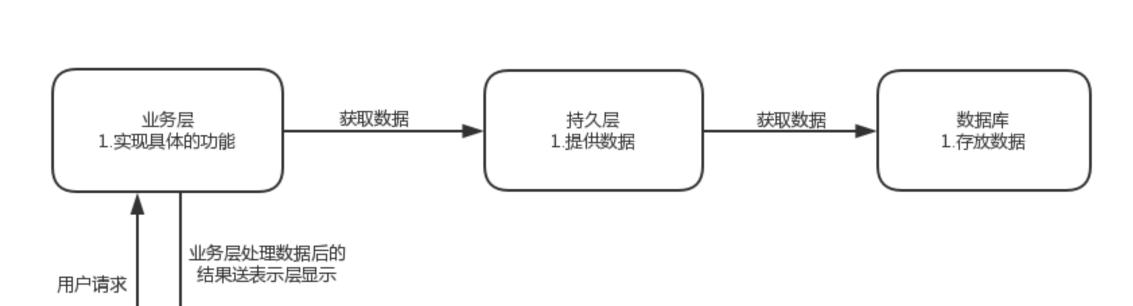
- ➤ Model 对应 Conference、User、Room等实体类;
- ➤ View 主要是 Web 页面形式,由 Vue.js 构建而成;
- ➤ Controller 主要包括 UserController.java、ConferenceController.java 等Java类。

MVC风格





- 系统分成四个层:表示层,业务层,持久层,数据库。
 - > 表示层: 用户界面、用户互动。将用户界面与具体功能隔离开。
 - ▶ 业务层:实现具体的的功能(会议管理、会议查询等)。
 - 持久层: 提供数据, 从数据库获取数据。
 - > 数据库: 存放数据, 会议室库、会议库等。



表示层 1.用户互动 2.显示数据

- ➤ 每一个URI代表一种资源;
- > 客户端和服务器之间, 传递这种资源的某种表现层;
 - 产在表现层之上,通过某种手段,让服务器端发生"状态转化"
- ➤ 客户端通过四个HTTP动词,对服务器端资源进行操作,实现"表现层状态转化"。
 - ▶ GET用来获取资源
 - > POST用来新建资源
 - > PUT用来更新资源
 - ➤ DELETE用来删除资源

GET /user/userID

接口描述

软件体系结构风格

根据token获取用户ID

—— RESTful风格

请求头

参数名	参数类型	描述	
Authorization	String	token	

响应(成功)参数

参数名	参数类型	描述	
status	Integer	0	
data	Integer	userID	

响应(失败)参数

参数名	参数类型	描述
status	Integer	-2
message	String	错误消息信息描述