

北京邮电大学



智能充电桩调度计费系统 面向对象设计文档

学院：计算机学院（国家示范性软件学院）

专业：计算机科学与技术

小组：14

编写者：张晨阳，梁维熙，廖轩毅，金建名

2025 年 5 月 29 号

版本修订记录					
编号	日期	版本号	章节	编写者	说明
1	2025-05-29	V1.0	2	梁维熙	完成部分用例实现方案设计——动态结构
			3	金建名	完成用例实现方案设计——静态结构
			2	张晨阳	完成部分用例实现方案设计——动态结构
			1	廖轩毅	完成软件体系结构设计

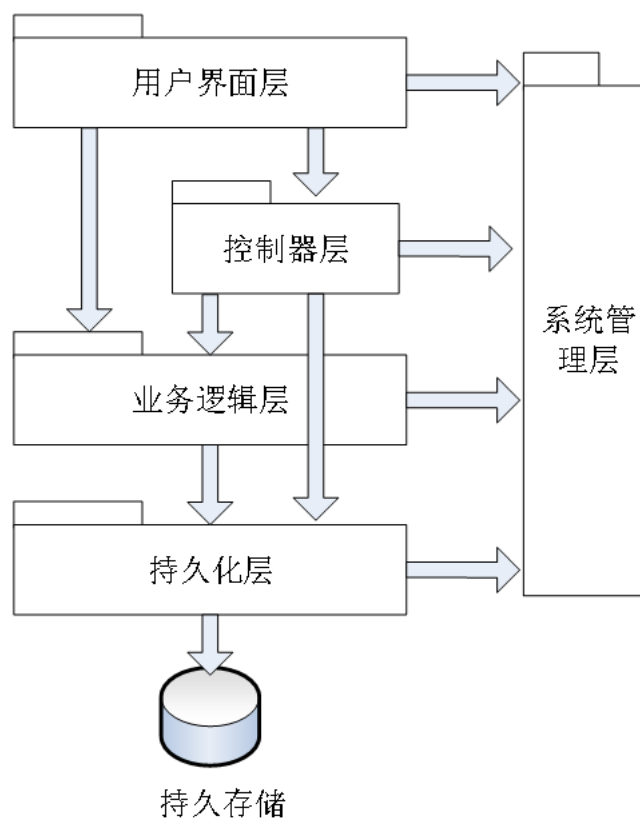
目录

1. 软件体系结构设计.....	1
2. 用例实现方案设计—创建动态结构.....	5
2.1. 充电申请	7
2.1.1. Modify_chargingRequest(car_Id, Request_Amount, Request_Mode) 7	
2.1.2. Query_Car_State(car_id).....	7
2.1.3. Query_Charging_State(car_id).....	8
2.1.4. End_Charging(car_id, ChargingPileNum)	8
2.2. 查看详细单	9
2.2.1. findByUserId(userId).....	9
2.2.2. findByChargingStationId(chargingStationId)	9
2.3. 运行充电桩	10
2.3.1. upsertChargingStation(ChargingStationUpsertRequest request) ...	10
2.3.2. upsertChargingStation(ChargingStationUpsertRequest request) ...	10
2.3.3. repairChargingStation(Long id).....	11
2.3.4. deleteChargingStation(Long id)	11
2.3.5. breakChargingStation(id)	12
2.4. 查看充电桩状态	13
2.4.1. getChargingStationWithSlot(Id).....	13
2.4.2. getAllChargingStationWithSlot()	13
2.5. 查看队列状态	14
2.5.1. getChargingStationWithSlot(Id).....	14
2.6. 用户信息维护	15
2.6.1. loginWithToken(UserRequest request).....	15
2.6.2. registerAndLogin(UserRequest request)	15
2.6.3. updateUsername(Long userId, String newUsername).....	16
2.6.4. updatePassword(Long userId, String oldPassword, String newPassword)	16

3. 用例实现方案设计—创建静态结构.....	17
3.1. 用户界面层设计类图	17
3.2. 控制器/处理层设计类图	18
3.3. 业务/领域层设计类图	19
3.4. 持久化层设计类图	19
3.5. 系统层设计类图	20

1.软件体系结构设计

本系统采用“分层”的体系结构，其软件分层结构如下所示：



用户界面层

组成：主要由各种用户交互组件构成，如图形界面中的窗口、按钮、文本框、下拉菜单等，对于 Web 应用则包括 HTML 页面、CSS 样式表以及用于增强交互性的 JavaScript 脚本等。

功能：

- **展示信息：**将系统处理后的数据以直观易懂的方式呈现给用户，如表格、图表、文本等形式展示查询结果、报表等。
- **接收用户输入：**收集用户的各种操作指令和数据输入，并将这些输入信息传递给下一层（控制器层）进行处理。
- **提供交互体验：**通过友好的界面设计和交互方式，让用户能够方便地与系统进行交互，提升用户体验，如响应式的布局、动画效果等来增强界面的友好性和吸引力。

控制器层

组成：通常是一些类或模块，它们主要负责协调用户界面层和业务逻辑层之间的交互，具体实现可能因开发框架和语言的不同而有所差异。

功能：

- **请求分发：**接收用户界面层传递过来的用户请求，根据请求的类型和内容，将请求分发到相应的业务逻辑层组件进行处理，起到一个中间调度的作用。
- **参数校验：**对用户输入的参数进行初步的校验，确保输入的数据格式、类型等符合业务逻辑层处理的要求，避免无效或错误的数据进入业务逻辑层，例如检查输入的日期格式是否正确、必填项是否已填写等。
- **结果反馈：**将业务逻辑层处理后的结果返回给用户界面层，以便展示给用户，同时也可以根据业务逻辑层的处理结果进行一些额外的操作，如设置页面的状态、显示提示信息等。

业务逻辑层

组成：包含了一系列的业务逻辑类或服务类，这些类封装了系统的业务规则和操作，通常会根据不同的业务功能划分成多个模块，例如用户管理模块、订单处理模块、充电桩管理模块等。

功能：

- **实现业务规则：**按照系统的业务需求，实现各种业务规则和流程，如处理用户的注册与登录请求、处理充电桩的状态变化请求等，是系统的核心功能部分，确保系统能够正确地处理各种业务场景。
- **数据处理与转换：**对从持久化层获取的数据进行处理和转换，以满足业务逻辑的需要，例如对数据进行排序、筛选、聚合等操作，同时也会将处理后的数据传递给持久化层进行存储或更新。
- **业务流程控制：**控制业务流程的执行顺序和逻辑，包括调用其他业务逻辑组件、处理异常情况等，确保业务流程的正确性和完整性，例如在处理一个复杂的业务流程时，需要按照一定的顺序依次调用多个业务逻辑方法，并在出现异常时进行相应的处理。

持久化层

组成：主要由数据访问对象（DAO）或数据访问层的类组成，这些类封装了对数据库或其他持久存储介质的访问操作。

功能：

- **数据存储与检索：**负责将业务逻辑层处理后的数据持久化到数据库或其他存储介质中，同时也从存储介质中检索数据并返回给业务逻辑层，确保数据的持久性和一致性，例如将用户信息存储到数据库表中，或者根据条件查询订单数据。
- **数据映射：**将业务逻辑层的对象模型与数据库的表结构或其他存储结构进行映射，实现对象与数据之间的转换，使得业务逻辑层可以使用面向对象的方式来操作数据，而无需直接处理底层的数据库操作，例如将一个用户对象映射到数据库中的用户表。
- **事务管理：**在需要的情况下，管理数据库事务，确保数据操作的原子性、一致性、隔离性和持久性，例如在进行多个相关数据的更新操作时，通过事务管理保证这些操作要么全部成功，要么全部失败，避免出现数据不一致的情况。

系统管理层

组成：包括系统配置管理模块、系统信息暂存模块、日志管理模块，这些模块共同构成了系统管理层，用于对整个软件系统的运行进行管理和维护。

功能：

- **系统配置管理：**负责管理系统的各种配置信息，如数据库连接配置、系统参数设置、用户权限配置等，这些配置信息可以方便地进行修改和更新，而无需修改代码，从而提高系统的灵活性和可维护性。
- **日志管理：**记录系统运行过程中的各种日志信息，包括用户的操作日志、系统的异常日志、性能日志等，便于对系统的运行情况进行跟踪和分析，帮助开发人员快速定位问题和进行系统优化。
- **安全认证：**实现对用户的身份认证和授权管理，确保只有合法的用户才能访问系统，并且用户只能访问其被授权的资源和功能，例如通过用户名和密码登录、使用令牌进行身份验证等。

持久存储

组成：主要是数据库系统（如关系型数据库 MySQL，非关系型数据库 Redis），也可能包括文件系统等其他存储介质，用于长期存储系统的数据。

功能：

- **数据持久化存储：**为系统提供稳定可靠的数据存储服务，将系统的重要数据（如用户信息、业务数据等）持久化存储起来，即使系统重启或出现故障，数据也不会丢失，确保数据的持久性和完整性。
- **数据查询与检索：**支持高效的查询和检索操作，使得系统能够快速地从存储介质中获取所需的数据，不同的存储介质提供了不同的查询机制和性能特点，例如关系型数据库通过 SQL 语句进行复杂的数据查询和关联操作，非关系型数据库则在处理海量数据和高并发场景下具有优势。
- **数据备份与恢复：**提供数据备份和恢复的功能，定期对数据进行备份，以便在数据丢失或损坏时能够快速恢复数据，确保系统的数据安全性和业务连续性。

2.用例实现方案设计—创建动态结构

表 1 系统操作表

用例	系统操作		返回信息
充电申请	1、提交/修改充电申请	Modify_chargingRequest(car_Id, Request_Amount, Request_Mode)	Return(car_position, car_state, queue_Num, request_time)
	2、查看队列状态	Query_Car_State(car_id)	Return(car_Number_before_position, car_state, queue_Num, request_time)
	3、查看充电状态	Query_Charging_State(car_id)	Return(详单信息, 具体参数详见需求说明)
	4、结束充电	End_Charging(car_id, ChargingPileNum)	Return(0/1)
查看详细单	1、用户查看详细单申请	findByUserId(userId)	Return(List(Order))
	2、管理员查看充电桩所有详单	findByChargingStationId(chargingStationId)	Return(List(Order))
运行充电桩	1、创建或修改充电桩状态	upsertChargingStation(ChargingStationUpsertRequest)	Return(ChargingStationResponse)
	2、维修充电桩	repairChargingStation(id)	Return(ChargingStationResponse)
	3、删除充电桩	deleteChargingStation(id)	无
	4、模拟充电桩故障	breakChargingStation(id)	Return(ChargingStationResponse)
查看	1、查看充电桩状态。要求定时刷新所有充电桩的状态并在	getChargingStationWithSlot(Id)	Return(id,name,mode,power,status,charge_queue,report)

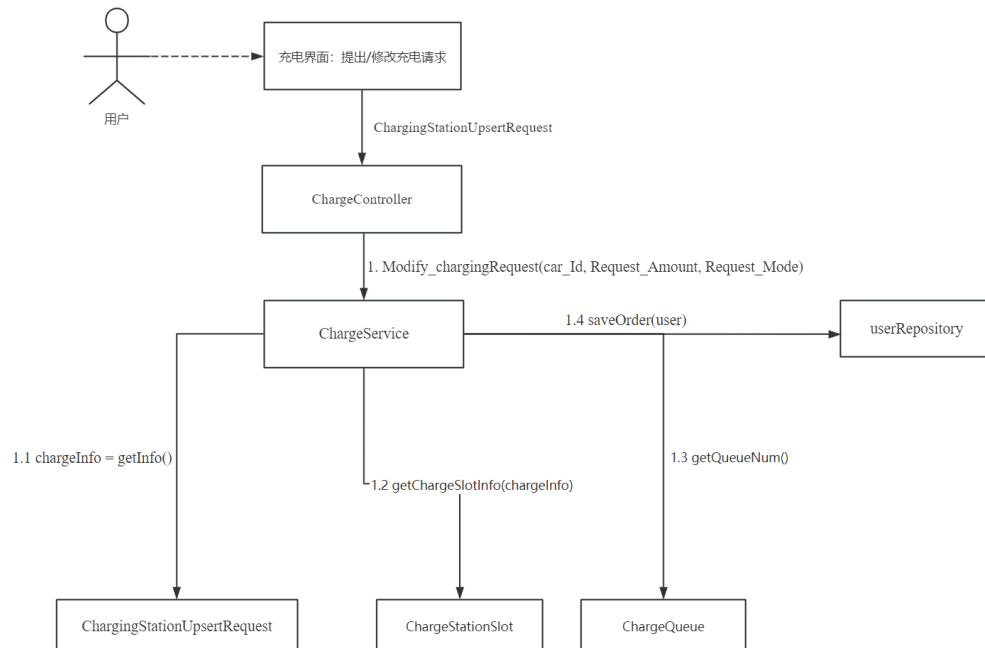
智能充电桩调度计费系统
面向对象设计文档

充电桩状态	客户端显示		
	2、获取所有充电桩的报表信息	getAllChargingStationWithSlot()	Return(List(id,name,mode,power,status,charge_queue,report))
查看队列状态	1、查看队列状态	getChargingStationWithSlot(Id)	Return(ChargingStationResponse)
用户操作	1、登录系统	loginWithToken(UserRequest)	Return(Token, UserResponse)
	2、注册用户	registerAndLogin(UserRequest)	Return(Token, UserResponse)
	3、修改用户名	updateUsername(userId,newUsername)	Return(UserResponse)
	4、修改密码	updatePassword(userId,oldPassword,newPassword)	无

2.1. 充电申请

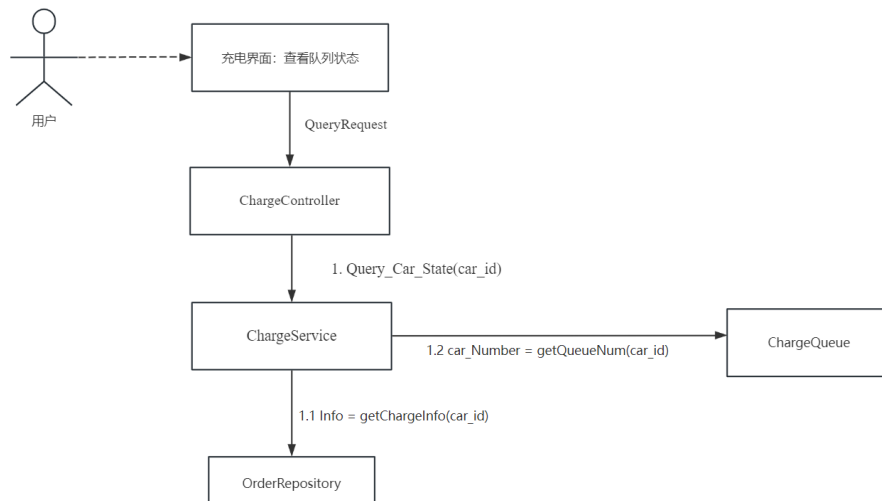
2.1.1. Modify_chargingRequest(car_Id, Request_Amount, Request_Mode)

协作图如下：



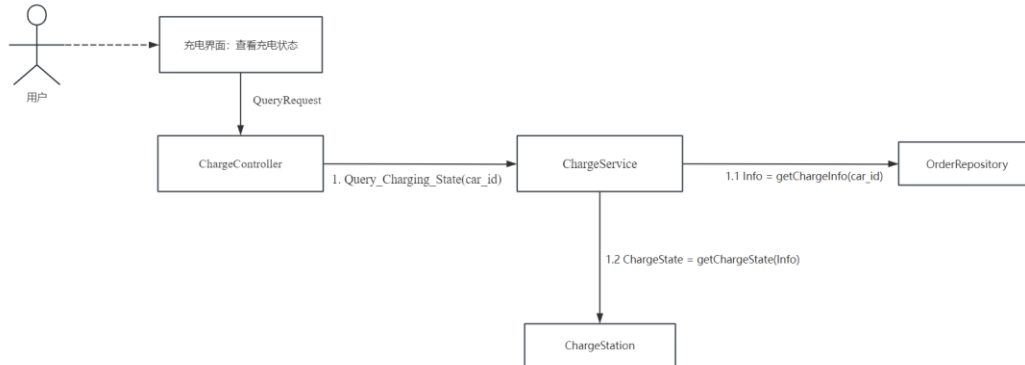
2.1.2. Query_Car_State(car_id)

协作图如下：



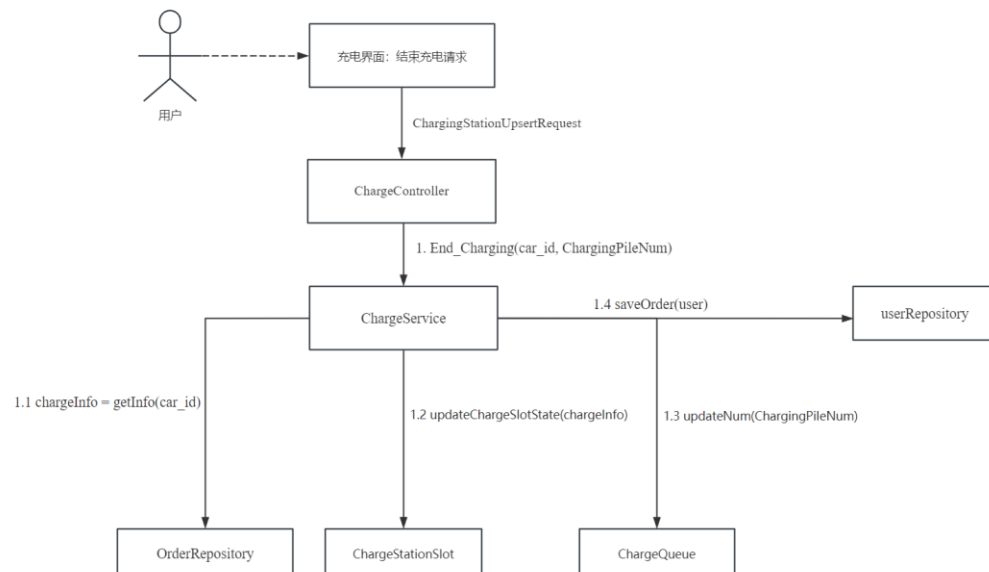
2.1.3. Query_Charging_State(car_id)

协作图如下：



2.1.4. End_Charging(car_id, ChargingPileNum)

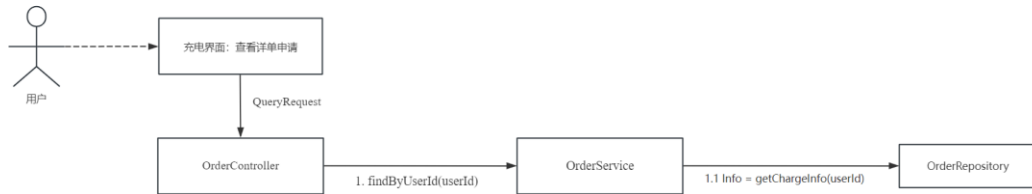
协作图如下：



2.2. 查看详情单

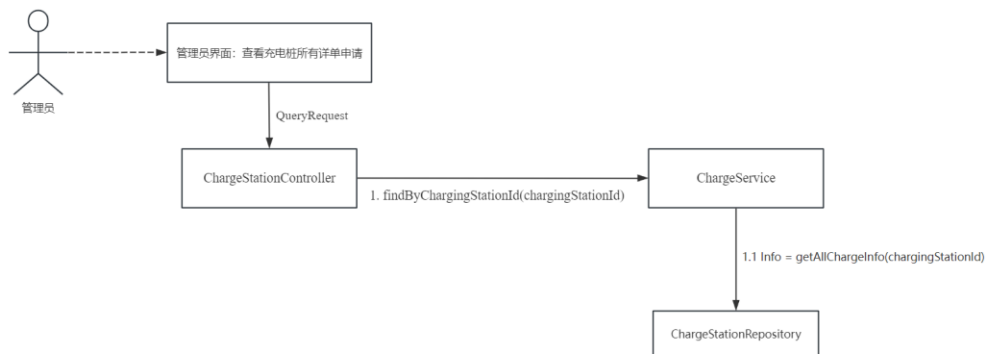
2.2.1. findByUserId(userId)

协作图如下：



2.2.2. findByChargingStationId(chargingStationId)

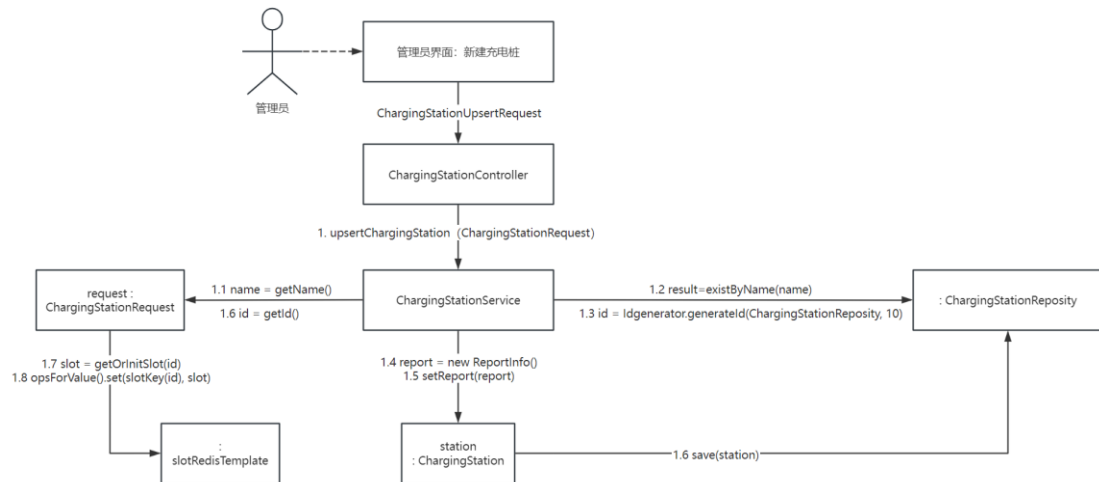
协作图如下：



2.3. 运行充电桩

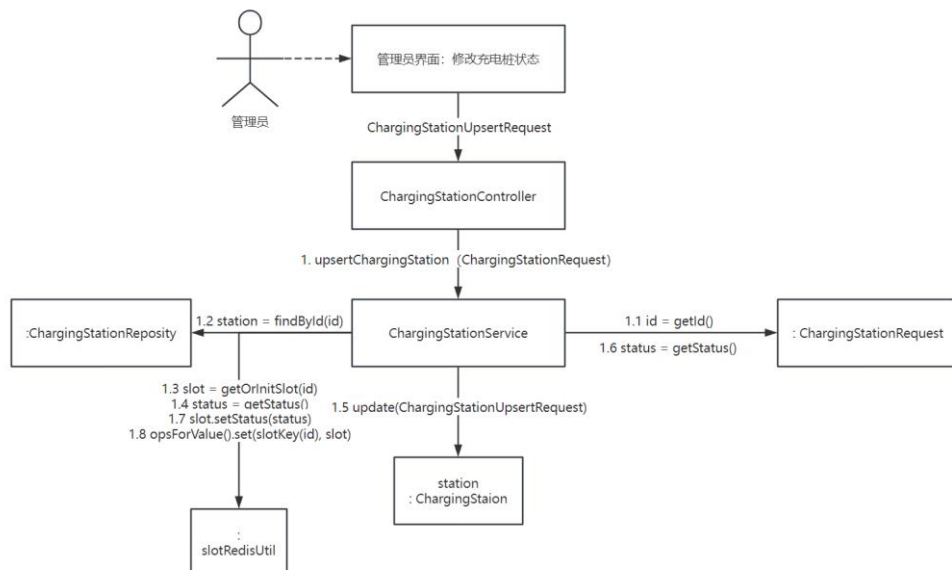
2.3.1. upsertChargingStation(ChargingStationUpsertRequest request)

协作图如下：



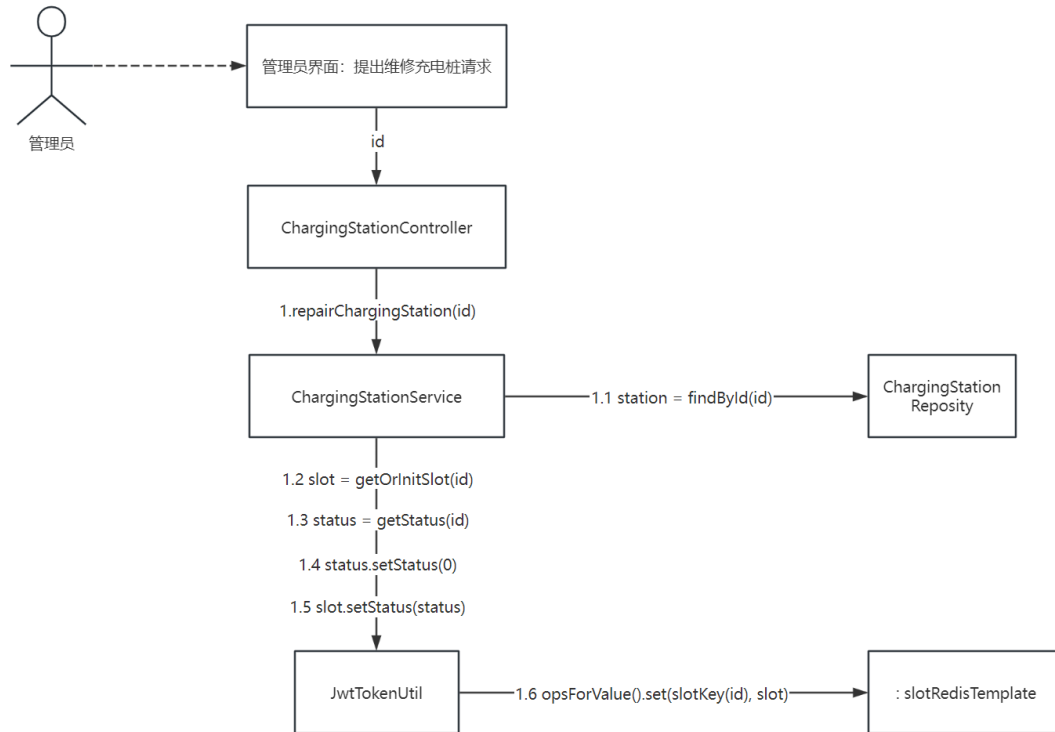
2.3.2. upsertChargingStation(ChargingStationUpsertRequest request)

协作图如下：



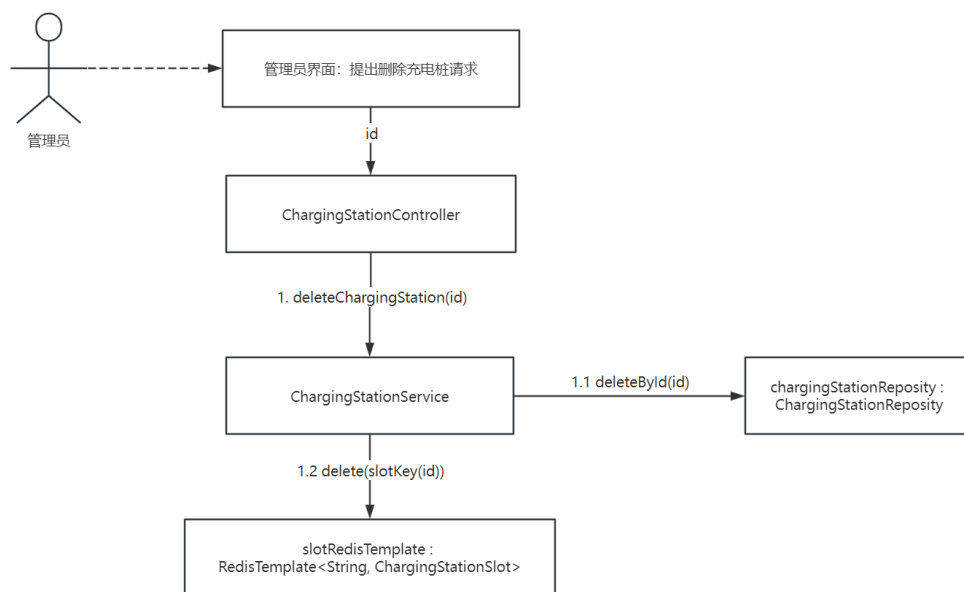
2.3.3. repairChargingStation(Long id)

协作图如下：



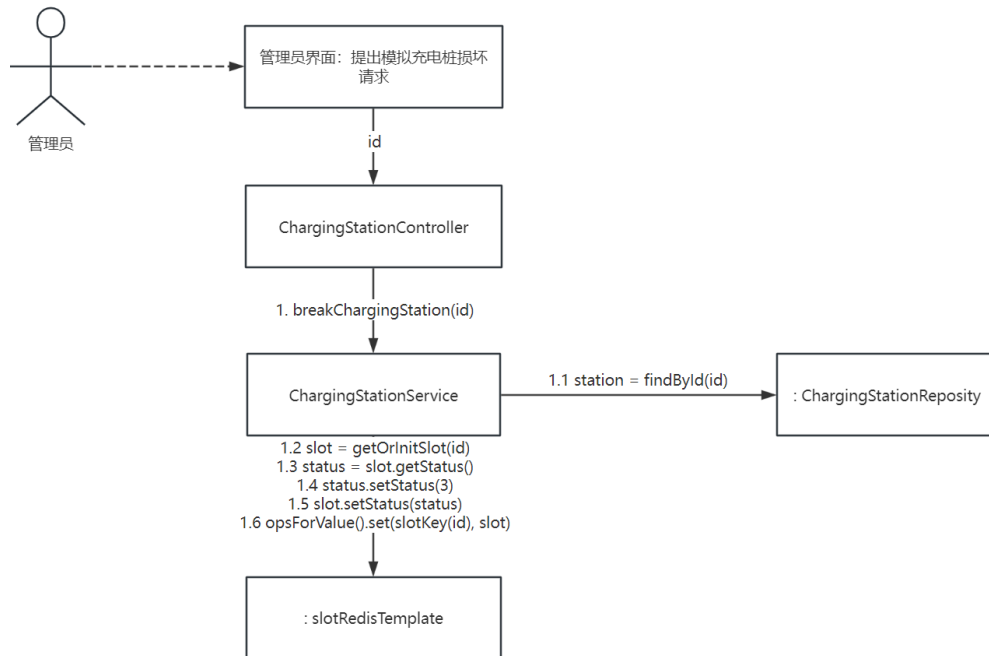
2.3.4. deleteChargingStation(Long id)

协作图如下：



2.3.5. breakChargingStation(id)

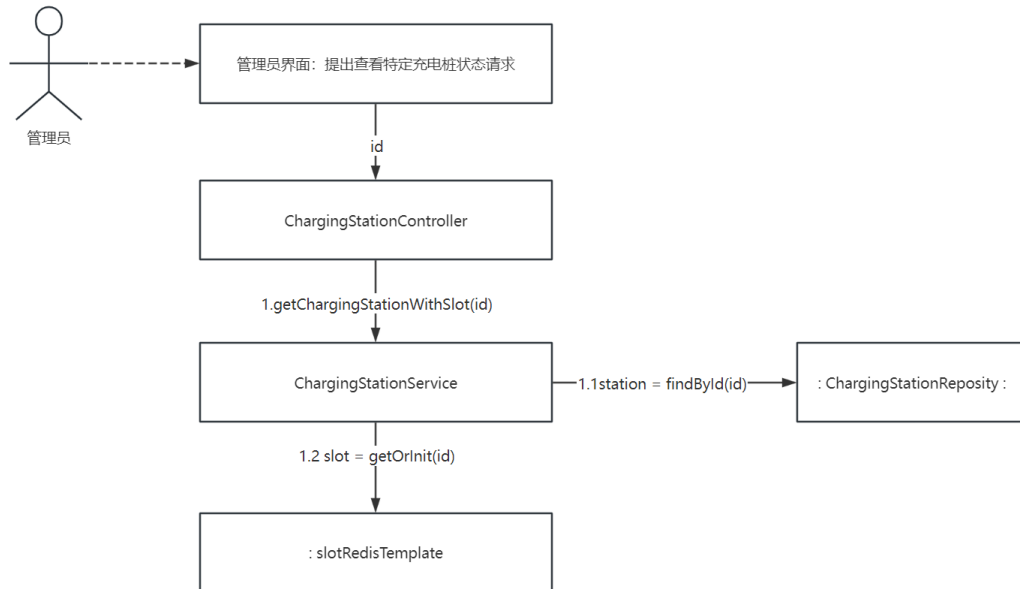
协作图如下：



2.4. 查看充电桩状态

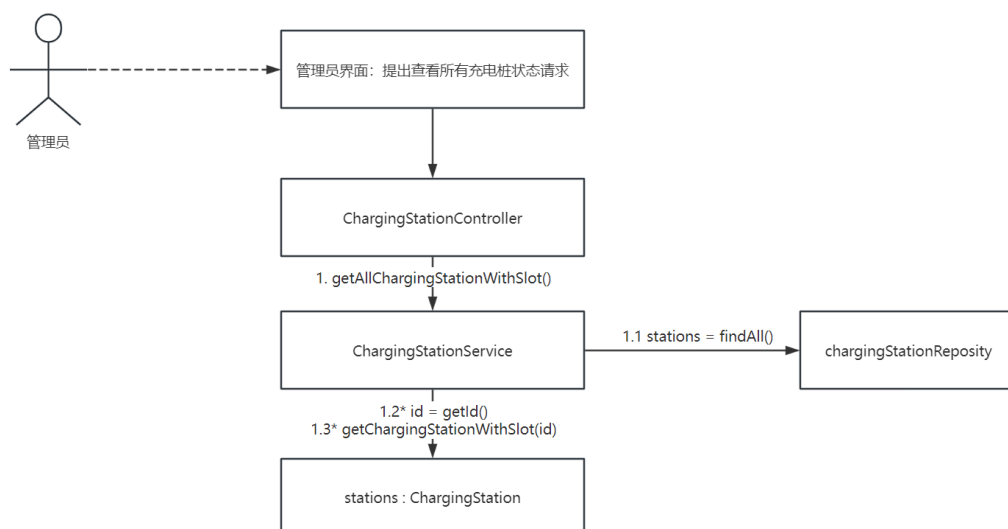
2.4.1. getChargingStationWithSlot(Id)

协作图如下：



2.4.2. getAllChargingStationWithSlot()

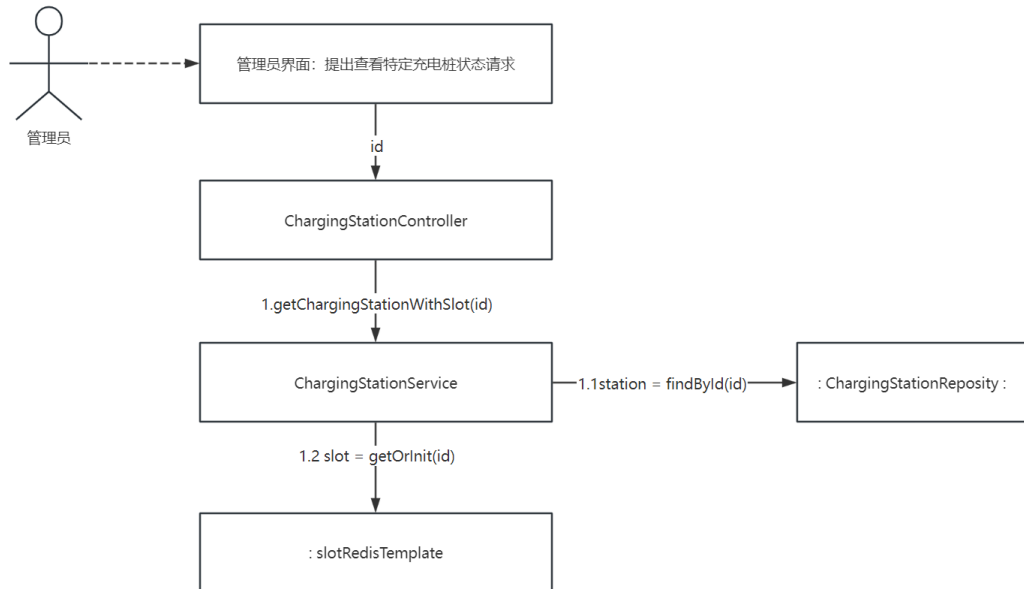
协作图如下：



2.5. 查看队列状态

2.5.1. getChargingStationWithSlot(Id)

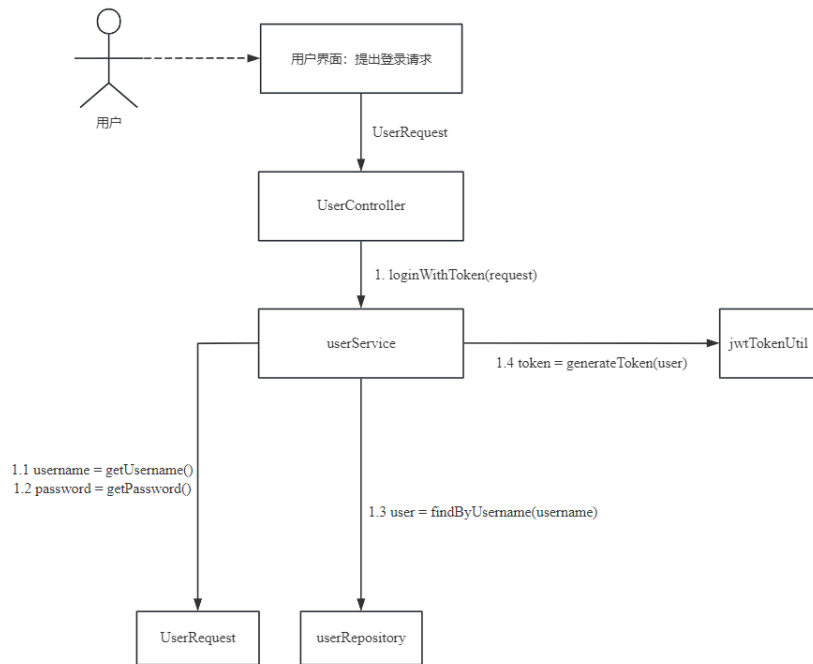
协作图如下：



2.6. 用户信息维护

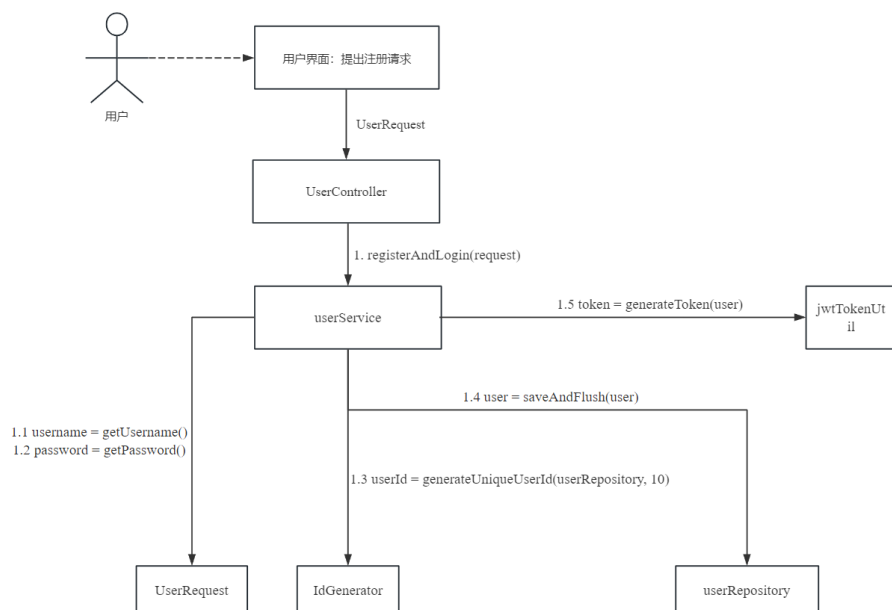
2.6.1. loginWithToken(UserRequest request)

协作图如下：



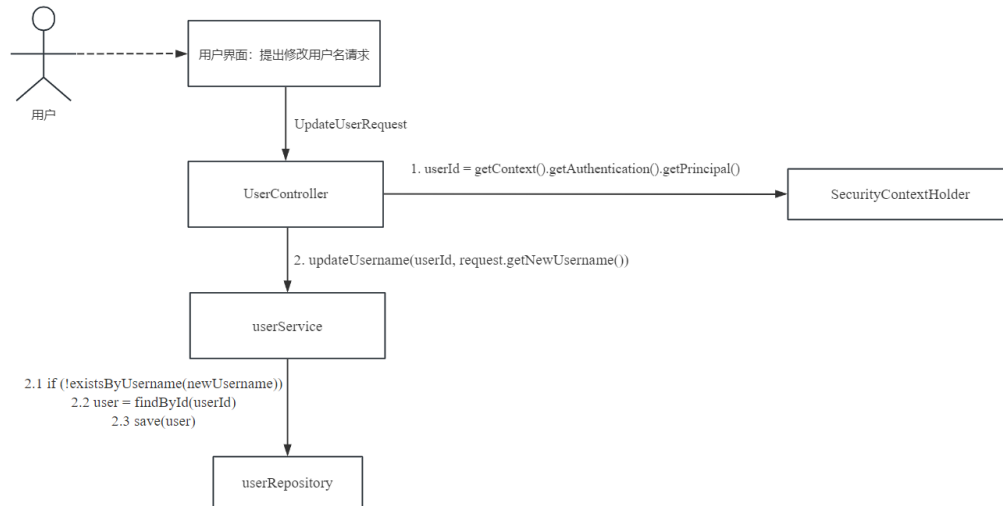
2.6.2. registerAndLogin(UserRequest request)

协作图如下：



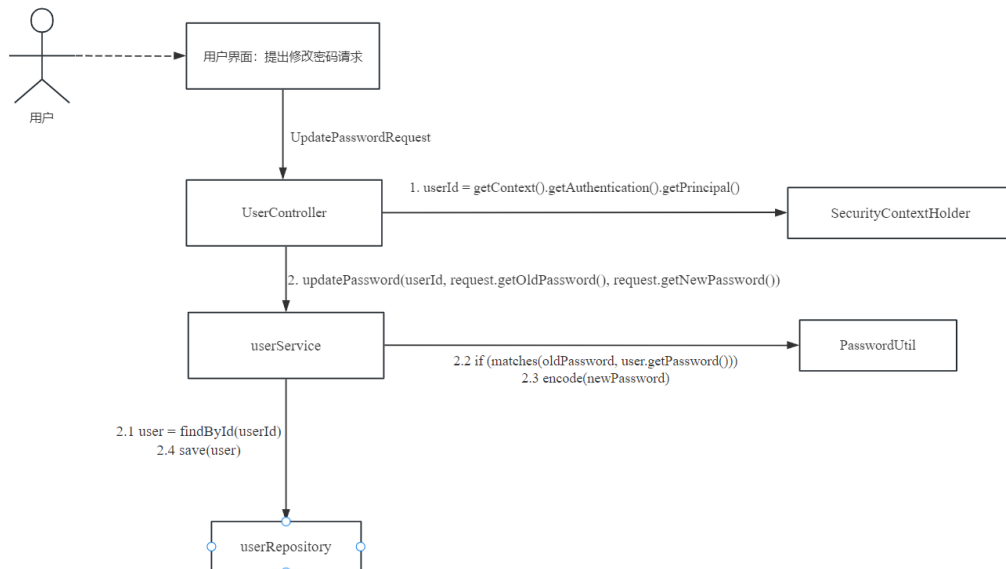
2.6.3. updateUsername(Long userId, String newUsername)

协作图如下：



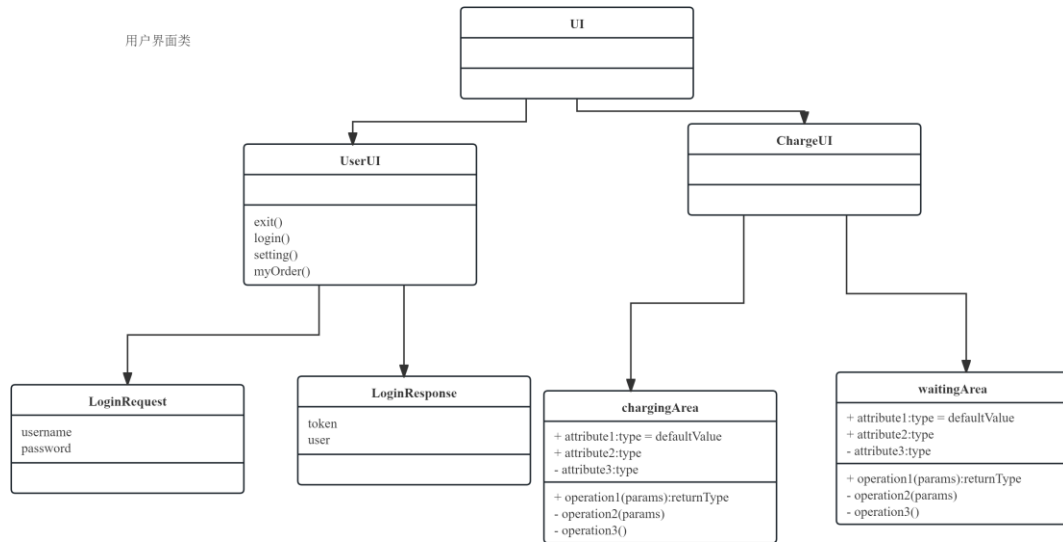
2.6.4. updatePassword(Long userId, String oldPassword, String newPassword)

协作图如下：



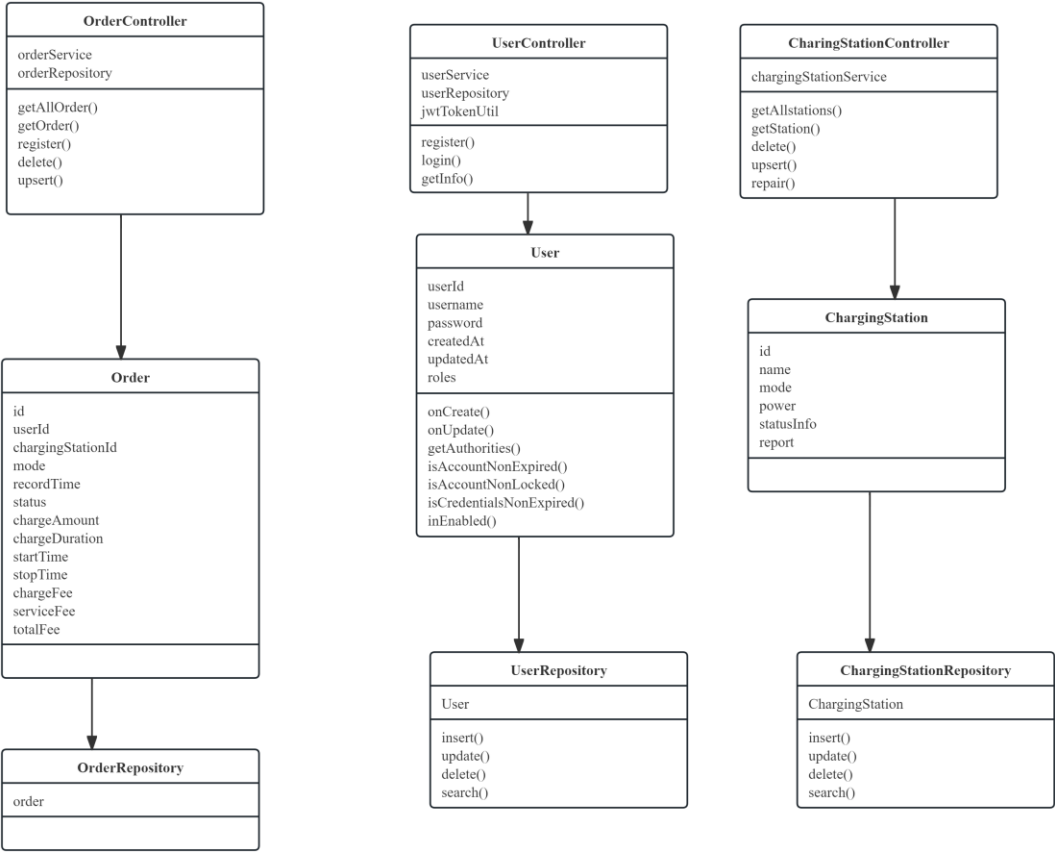
3.用例实现方案设计—创建静态结构

3.1. 用户界面层设计类图

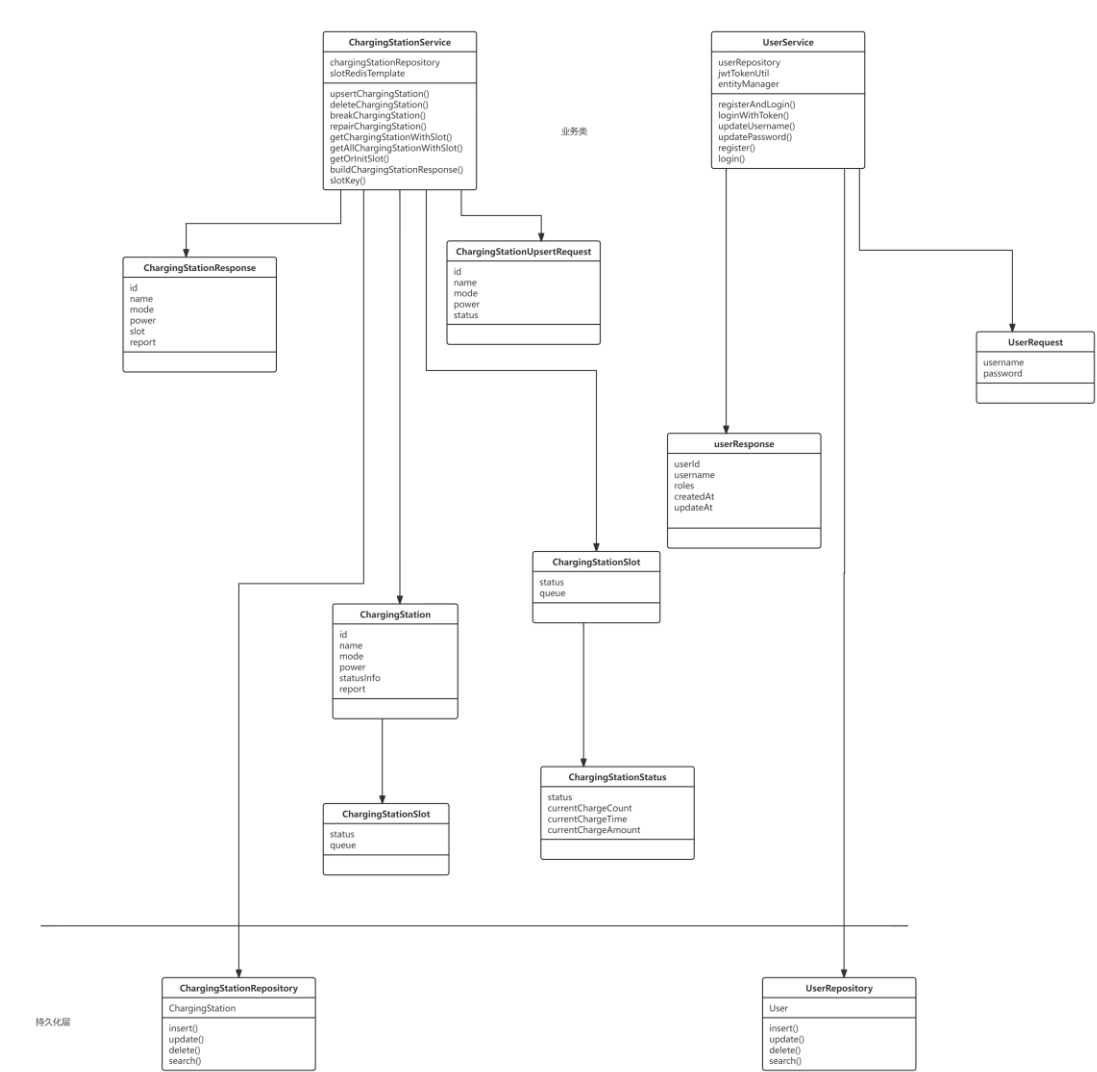


3.2. 控制器/处理层设计类图

控制器类



3.3. 业务/领域层设计类图



3.4. 持久化层设计类图



3.5. 系统层设计类图

JwtAuthenticationFilter
jwtTokenUtil
doFilterInternal()

OpenApiConfig
customOpenAPI()

Redisconfig
redisTemplate()

RegisterResquest
username password

ReportInfo
totalChargeCount toalChargeTime totalChargeAmount totalChargeFee totalServiceFee totalFee

SecurityConfig
jwtSecret jwtTokenUtil
securityFilterChain() jwtAuthenticationFilter() passwordEncode()

standardResponse
code msg data
success() error() StandardResponse()