rBook详细设计规约

修订历史：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编写日期 | 版本 | 说明 | SPEG | 修订人 |
| 2018.12.26 | 1 | 初始编写 | 软件工程课程杜庆峰老师 | 俞杨 |
| 2018.1.6 | 2 | 添加客户端相关类的详细设计 | 同上 | 张嘉洛 |

## 引言

* 1. 编写目的

本规约系在概要设计规约基础上对类设计进一步的细化，明确其属性、方法及协作关系，以作为代码生成的依据。

* 1. 编写依据

[rBook概要设计规约](rBook概要设计规约.docx)

* 1. 参考资料

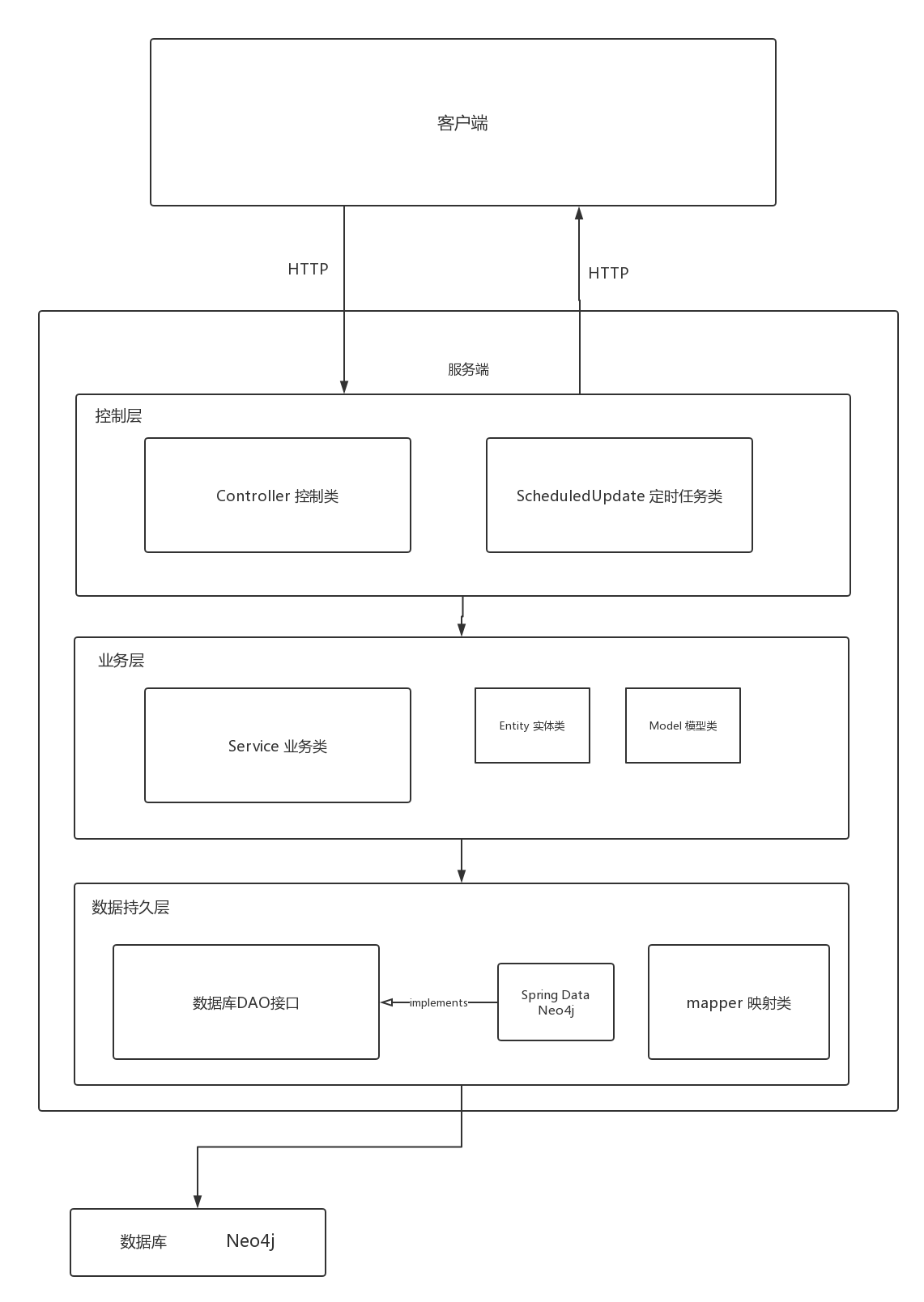
[Spring Boot Reference Document](https://docs.spring.io/spring-boot/docs/2.1.1.RELEASE/reference/htmlsingle/)

[Spring Data Neo4j Reference Document](https://docs.spring.io/spring-data/neo4j/docs/5.1.3.RELEASE/reference/html/)

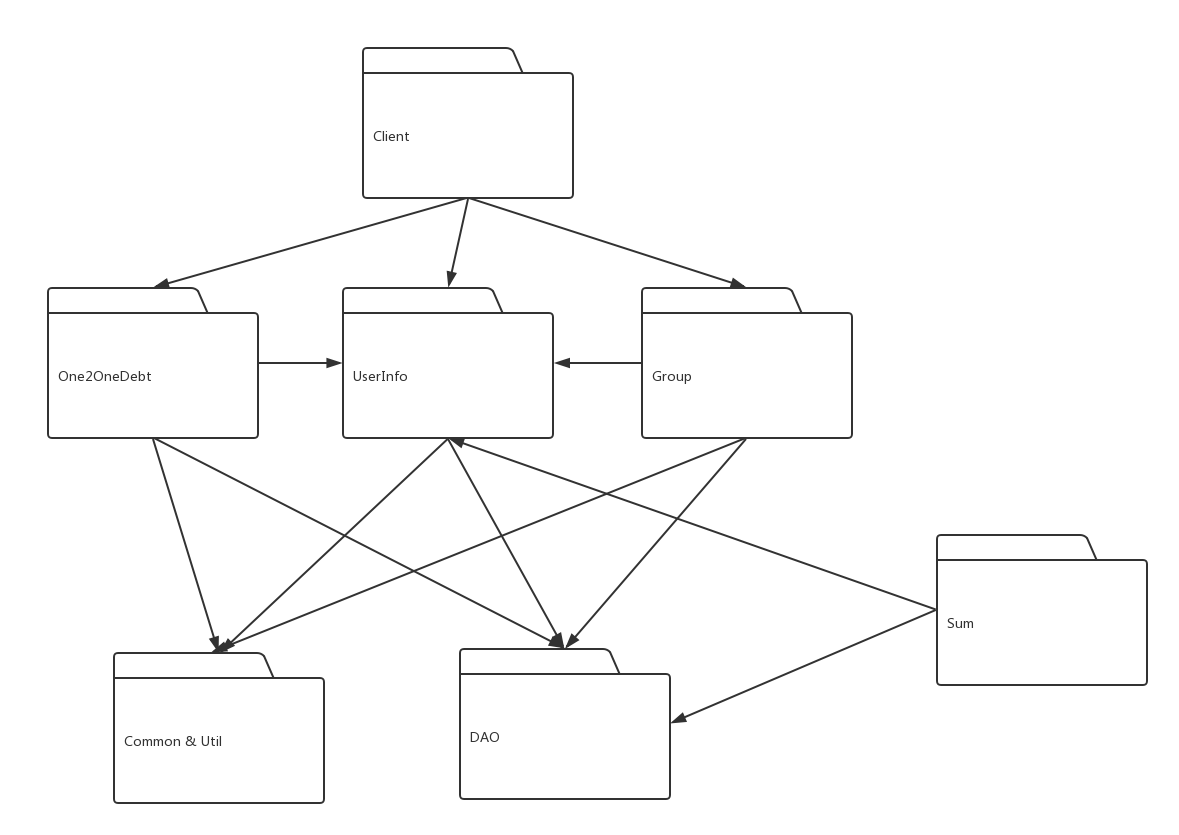
## 系统软件结构

同概要设计规约

系统层次架构如下：

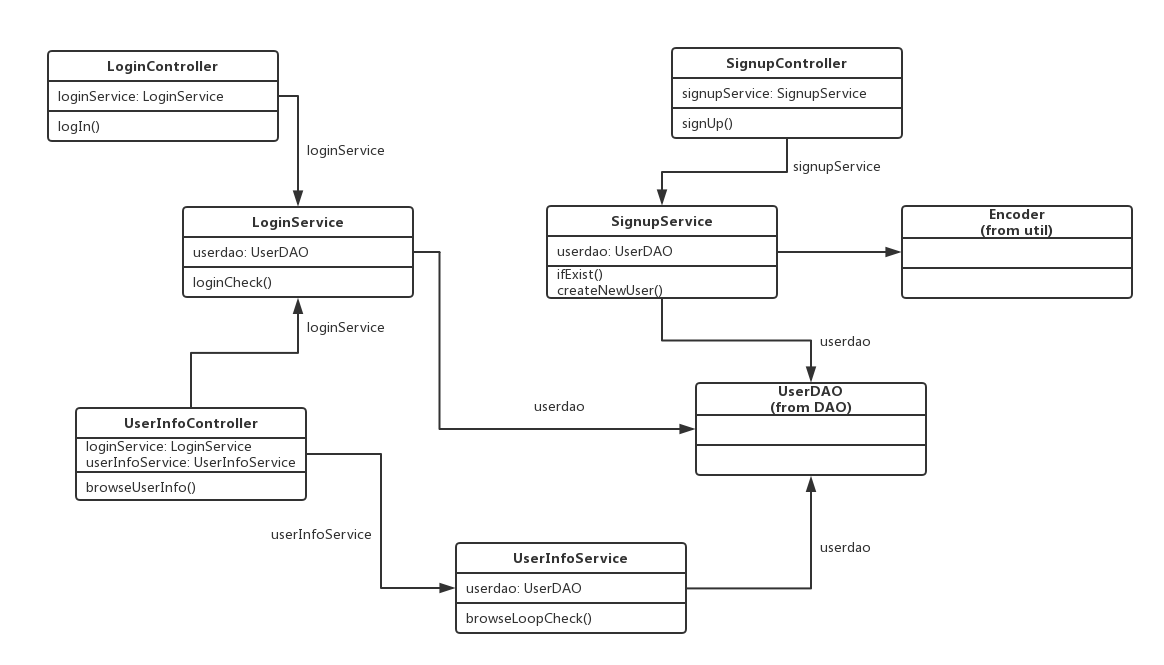


子系统架构如下：



## 类设计

* 1. **User相关**



3.1.1 LoginController类

说明：接受登陆HTTP请求并在业务处理完毕后给予回复的控制类

属性：

loginService: LoginService 通过Spring Boot注解依赖注入的业务类

方法：

logIn(req: LoginRequest): IfSuccessResponse 接受HTTP请求，调用loginService的loginCheck()方法进行登陆业务处理，并在完成后给予回复，返回IfSuccessResponse响应类

3.1.2 LoginService类

说明：进行登陆业务处理的服务类

属性：

userdao: UserDAO 通过Spring Boot 注解依赖注入的数据库交互类

方法：

loginCheck(username: String, password: String): User 进行用户登陆验证的业务方法，调用UserDao的 loginCheck()方法访问数据库用户信息来进行业务处理，返回登录成功的用户User类

3.1.3 SignupController类

说明：接受注册HTTP请求并在业务处理完毕后给予回复的控制类

属性：

signupService: SignupService 通过 Spring Boot 注解依赖注入的业务类

方法：

signUp(req: SignupRequest): IfSuccessResponse 接受HTTP请求，调用signupService的ifExist()方法判断用户是否存在，根据结果调用signupService的createNewUser()方法创建新用户，返回IfSuccessResponse响应类

3.1.4 SignupService类

说明：进行注册业务处理的服务类

属性：

userdao: UserDAO通过Spring Boot 注解依赖注入的数据库交互类

方法：

ifExist(name: String, identity: String): Boolean 判断目标用户名及身份信息在系统中是否存在，调用userDAO的checkExistList()方法访问数据库信息来进行业务处理，返回判断结果

createNewUser(username: String, password: String, identity: String, nickname: String): Boolean 创建指定信息的用户，调用Encoder类的静态方法encryptBasedDes()来对密码信息加密后，调用userDAO的addUser()方法向数据库写入信息，返回注册是否成功的结果boolean

3.1.5 UserInfoController类

说明：接受查看用户信息HTTP请求并在业务处理完毕后给予回复的控制类

属性：

loginService: LoginService 通过Spring Boot注解依赖注入的业务类，用于在业务前进行权限检查

userInfoService: UserInfoService 通过Spring Boot注解依赖注入的业务类

方法：

browseUserInfo(req: BrowseRequest): IfSuccessResponse 接受HTTP请求，调用loginService的loginCheck()方法检查权限后，调用userInfoService的browseLoopCheck()方法获取环路信息，返回IfSuccessResponse响应类

3.1.6 UserInfoService 类

说明：进行查看用户信息的业务类（由于其他用户信息在登录时就已经返回，该类暂只用于查看环路信息）

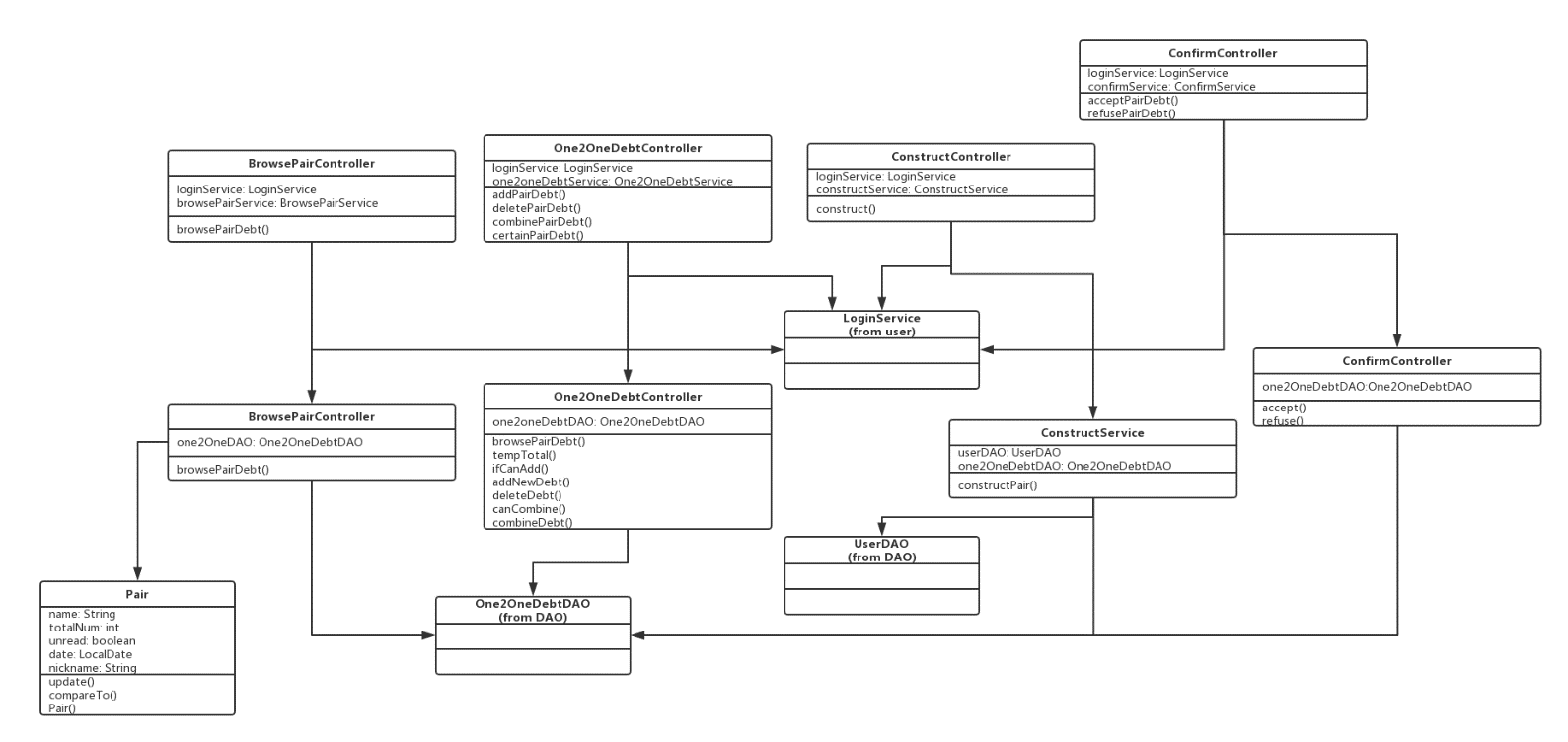
属性：

userDAO: UserDAO 通过Spring Boot 注解依赖注入的数据库交互类

方法：

browseLoopCheck(username: String): List<LoopCheck> 调用userDAO的 getLoopCheck()方法与数据库交互访问用户环路信息，返回LoopCheck环路类列表

* 1. **One2OneDebt相关**



* + 1. BrowsePairController 类

说明：接受查看用户结对列表HTTP请求并在业务处理完毕后给予回复的控制类

属性：

loginService: LoginService 通过Spring Boot注解依赖注入的业务类，用于在业务前进行权限检查

browsePairService: BrowsePairService 通过Spring Boot注解依赖注入的业务类

方法：

browsePairDebt(req:BrowseReques): IfSuccessResponse 接受HTTP请求，调用loginService的loginCheck()方法检查权限后，调用browsePairService的browsePair()方法获取结对信息，返回IfSuccessResponse响应类

* + 1. BrowsePairService 类

说明：进行查看用户所处结对关系列表的业务类

属性：

one2OneDAO: One2OneDebtDAO 通过Spring Boot 注解依赖注入的数据库交互类

方法：

browsePair(username: String): Map<String,Pair> 调用one2OneDAO的findAllDebts()方法访问数据库获取用户的所有一对一账目信息One2OneDebt后，处理封装成用于浏览结对关系的Pair类，返回结对关系对方用户名和Pair类的键值对Map<String,Pair>

* + 1. Pair类

说明：适用于浏览用户结对关系的Model类，是对One2OneDebt实体类的组装

属性：

name: String 结对关系中对方用户名

totalNum: int 总金额，正代表资金从用户流出到对方

unread: Boolean 新操作表示，true代表存在未被确认的账目

date: LocalDate 关系下最新一笔账目的时间

nickname: String 对方的昵称

方法：

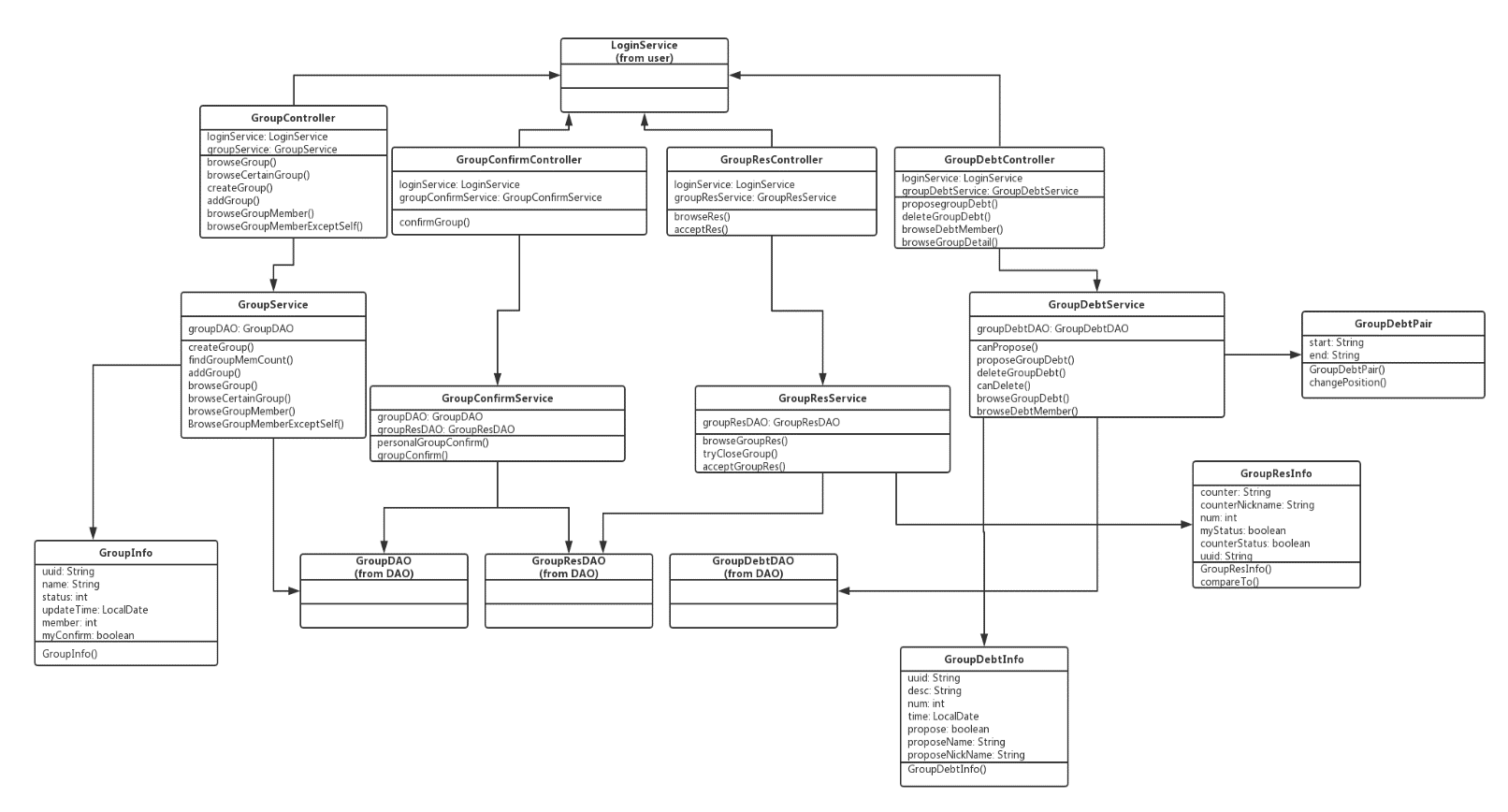
update(num: int, out: boolean, date: LocalDate, unread: boolean): void 用于封装Pair时更新信息的方法，根据输入的账目属性更新自身状态

compareTo(o: Pair): int：重写Object的比较，用于排序，使得存在新操作和时间较新的Pair能在使用Collections.sort（）方法排序时排在前端

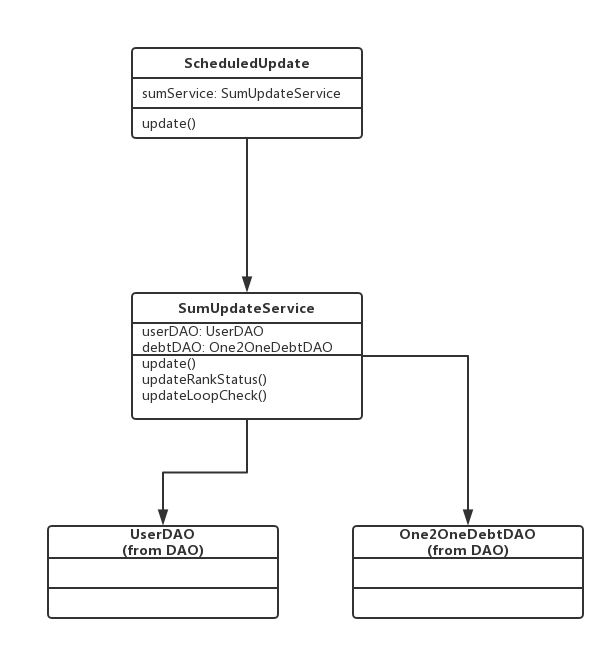
Pair(name: String, totalNum: int, unread: boolean, date: LocalDate, nickname: String): 以属性为参数的构造函数

3.2.4

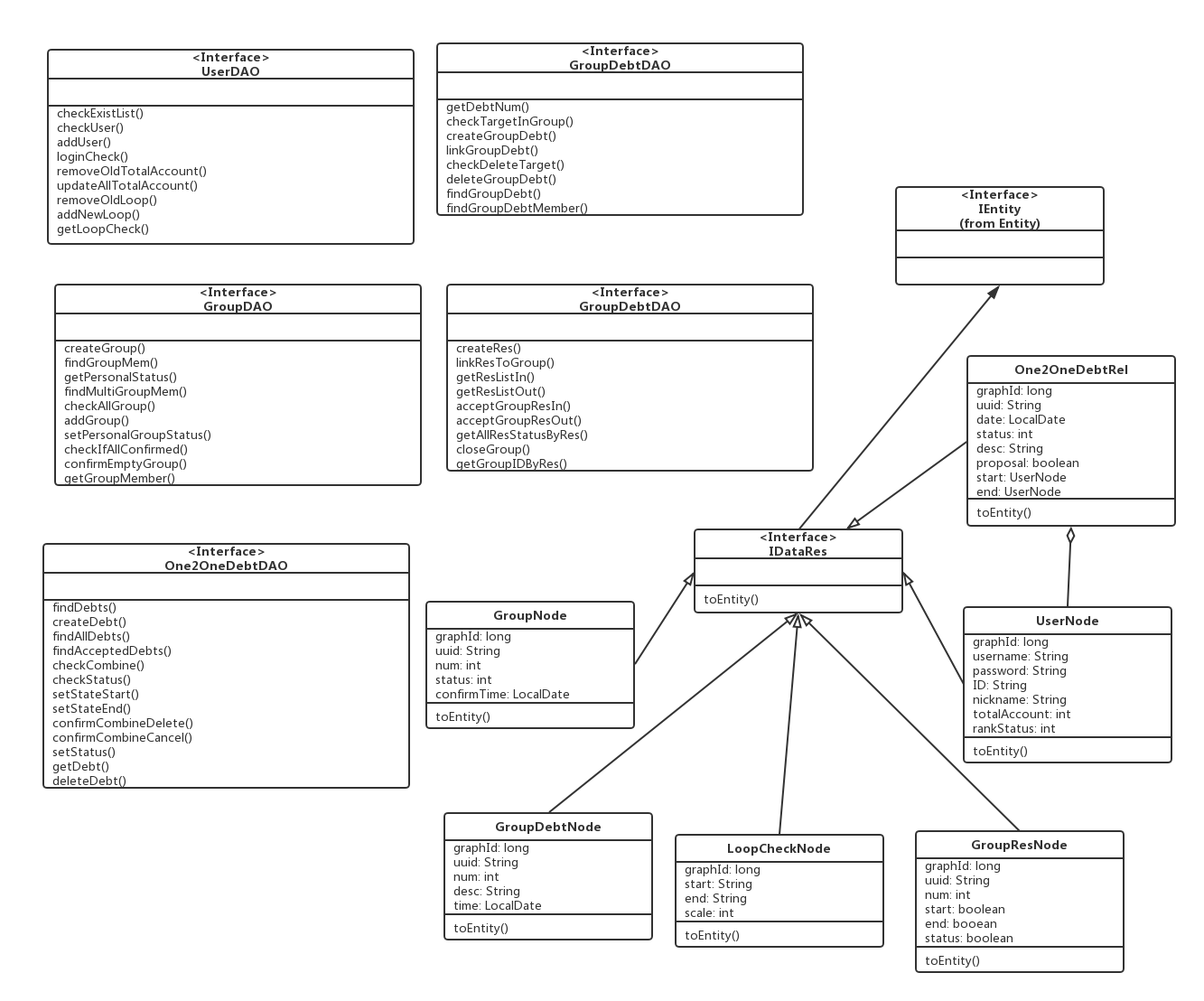
* 1. **Group相关**



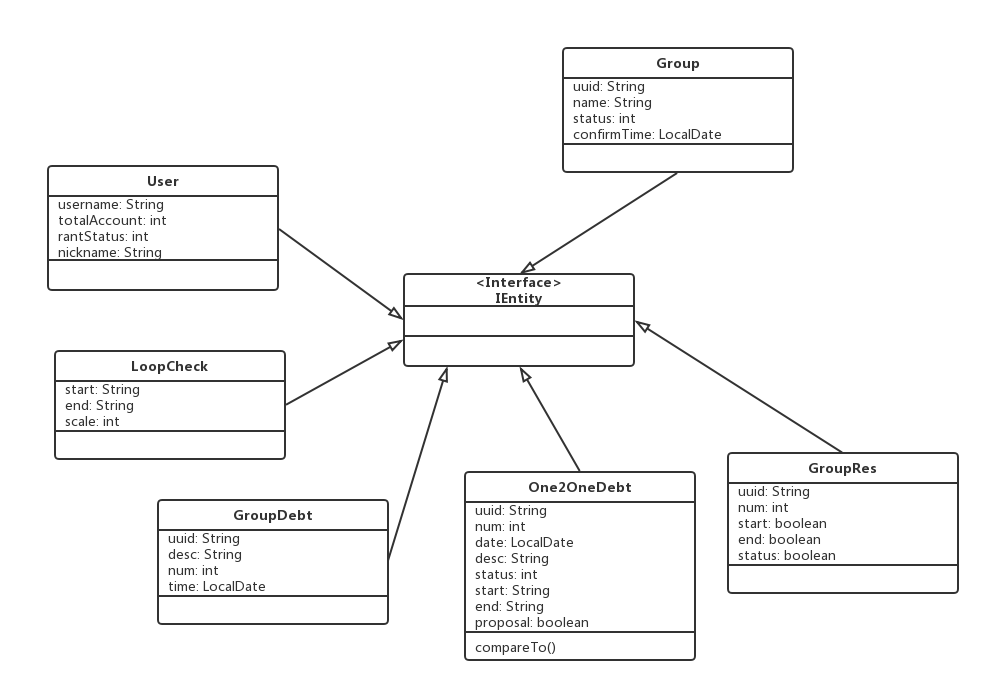
* 1. **Sum相关**



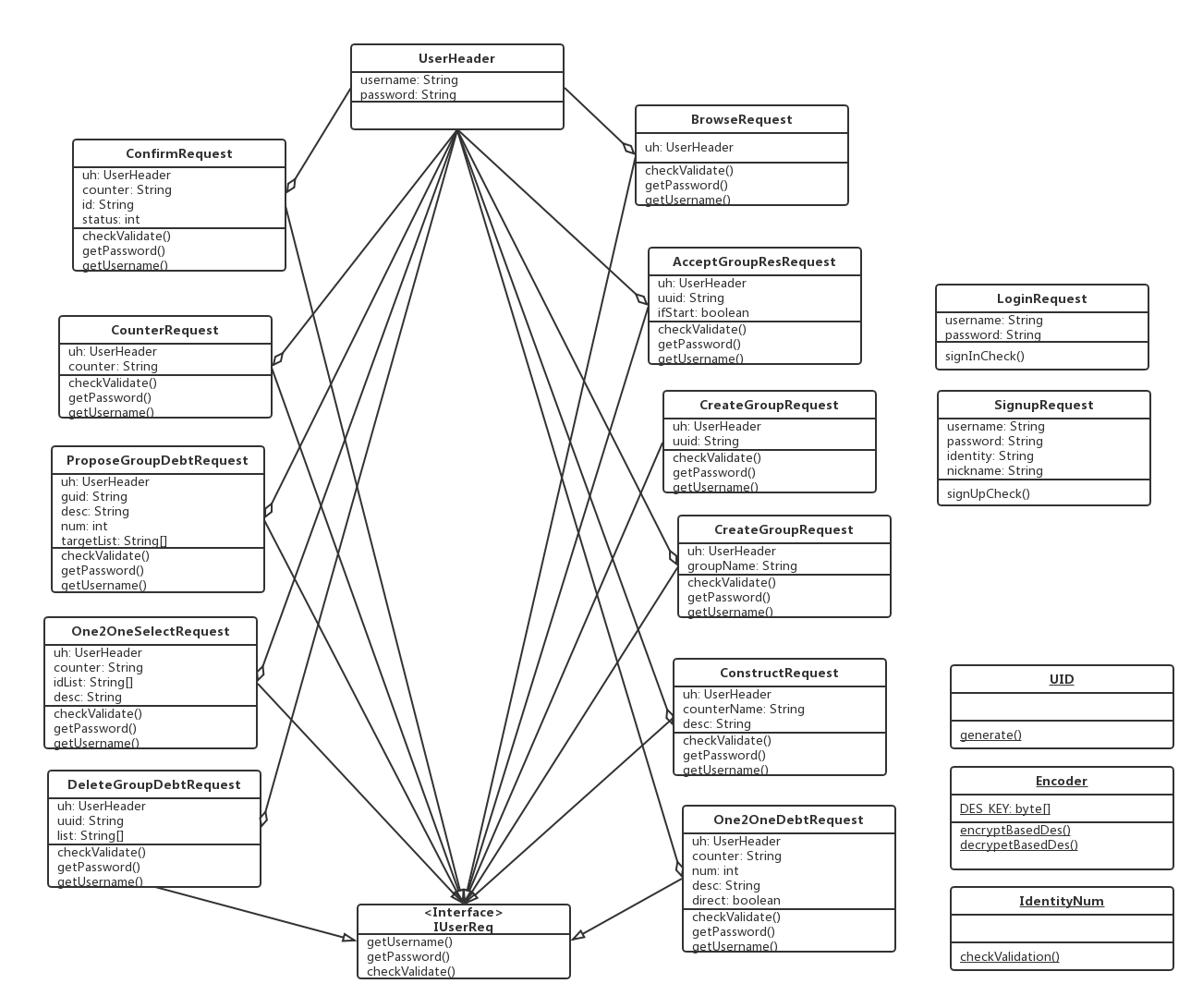
* 1. **DAO相关**



* 1. **Entity相关**



* 1. **Common & Util相关**



**3.8 Client相关**