ERP系统软件体系结构描述文档

小组分工

学号	完成部分
201250208	添加数据层职责，添加数据层接口，修改数据库逻辑层，添加接口格式
201250209	模块职责，层间接口，修改销售模块用例图
201250210	组合视角、销售模块接口规范
201250203	文档框架、引言、产品概述、逻辑视角、库存模块、信息视角

1.引言

1.1编制目的

本报告详细完成对ERP系统（企业资源管理系统）的概要设计，达到指导详细设计和开发的目的，同时实现和测试人员及用户的沟通。

本报告面向开发人员、测试人员及最终用户而编写，是了解系统的导航。

1.2词汇表

词汇名称	词汇含义	备注
ERP	企业资源管理系统	……

1.3参考资料

软件开发的技术基础 附录D.3

2.产品概述

公司规模扩大，企业业务量、办公场所、员工数都发生增长，为适应新的环境，提高 工作效率和用户满意度，现制作此产品，要求大幅增加销售额 ,提高销售人员的工作效率，能够清晰展现销售人员业绩。

3.逻辑视角

ERP系统中，选择了分层体系结构风格，将系统分为3层（展示层、业务逻辑层、数据层）能够很好地示意整个高层抽象。展示层包含GUI页面的实现，业务逻辑包含业务逻辑处理的实现，数据层负责数据的持久化和访问。分层体系结构的逻辑视角和逻辑设计方案如图1和图2所示。

图1 参照体系结构风格的包图表达逻辑视角

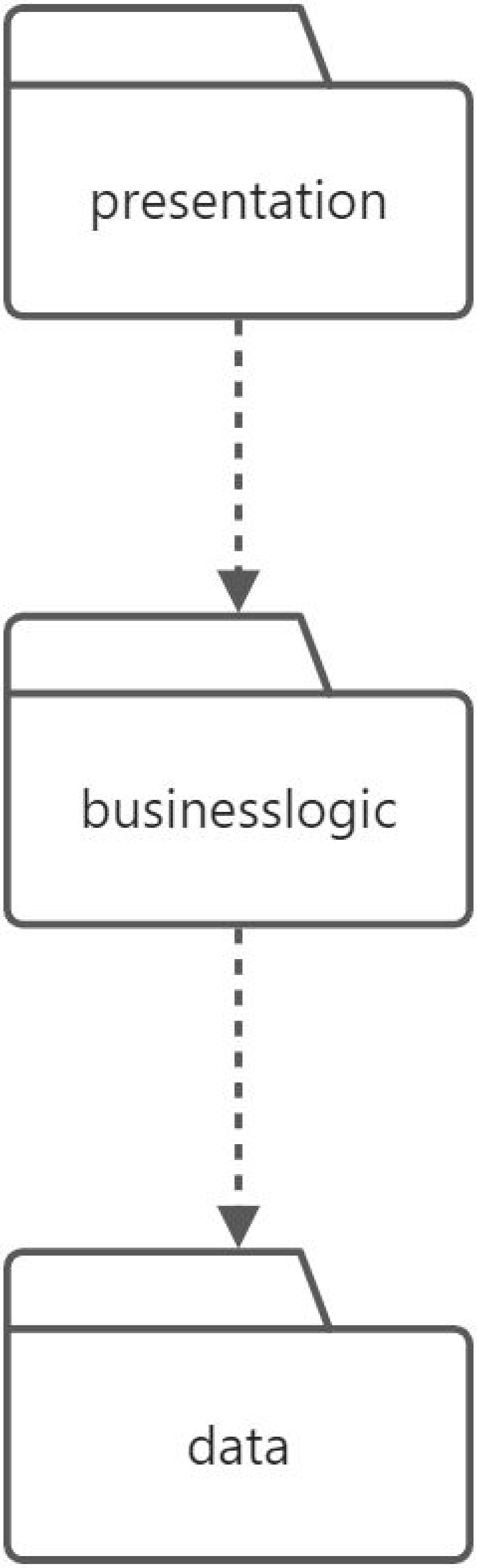
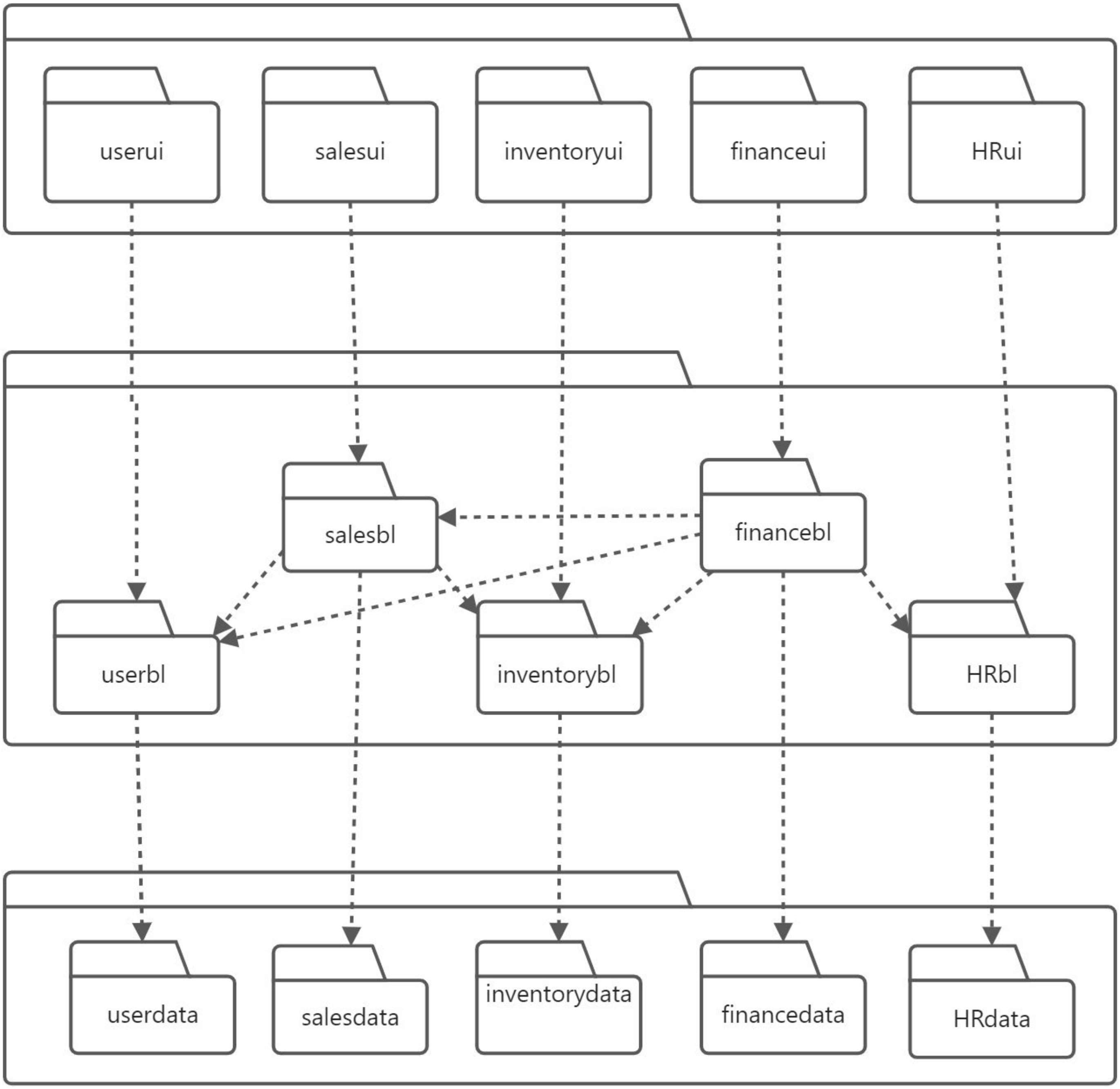


图2 软件体系结构逻辑设计方案



4.组合视角

4.1开发包图

ERP系统的最终开发包设计如表1所示。

表1 ERP系统的最终开发包设计

开发包	依赖的其他开发包
userui	usercontroller, vue类库包, vo,view api

saleui	salecontroller,vue类库包, vo,view api
inventoryui	inventorycontroller,vue类库包, vo,view api
financeui	financecontroller,vue类库包, vo,view api
hrui	hrcontroller,vue类库包, vo,view api
userservice	userdata
usercontroller	userservice
salesservice	userservice,inventoryservice,salesdata
salescontroller	salesevice
inventoryservice	financeservice,inventorydata
inventorycontroller	inventoryservice
financeservice	financedata,inventoryservice, salesservice
financecontroller	financeservice
hrservice	hrdata
hrcontroller	hrservice
userdata	po,Databaseutibility
usermapper	po,userdata
salesdata	po,Databaseutibility
salesmapper	po,salesdata
inventorydata	po, Databaseutibility
inventorymapper	po,inventorydata

financedata	po,Databaseutibility
financemapper	po,financedata
hrdata	po,Databaseutibility
hrmapper	hrdata, po
vue类库包	
databaseutility	JDBC,mysql
java RMI	
po	
vo	
view api	usercontroller, salescontroller, inventorycontroller, financecontroller, hrcontroller

ERP系统前端开发包图如图3所示，后端开发包图如图4所示。

图3 ERP系统客户端开发包图

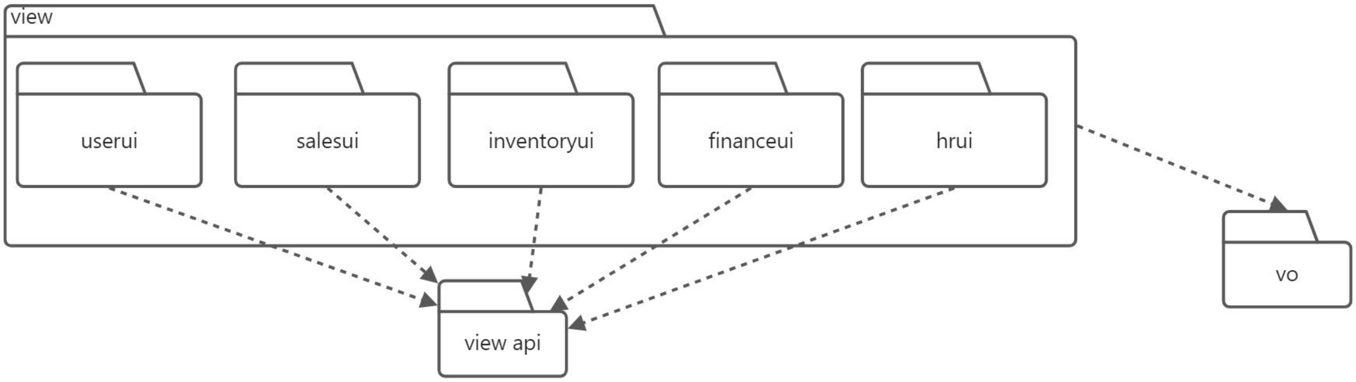
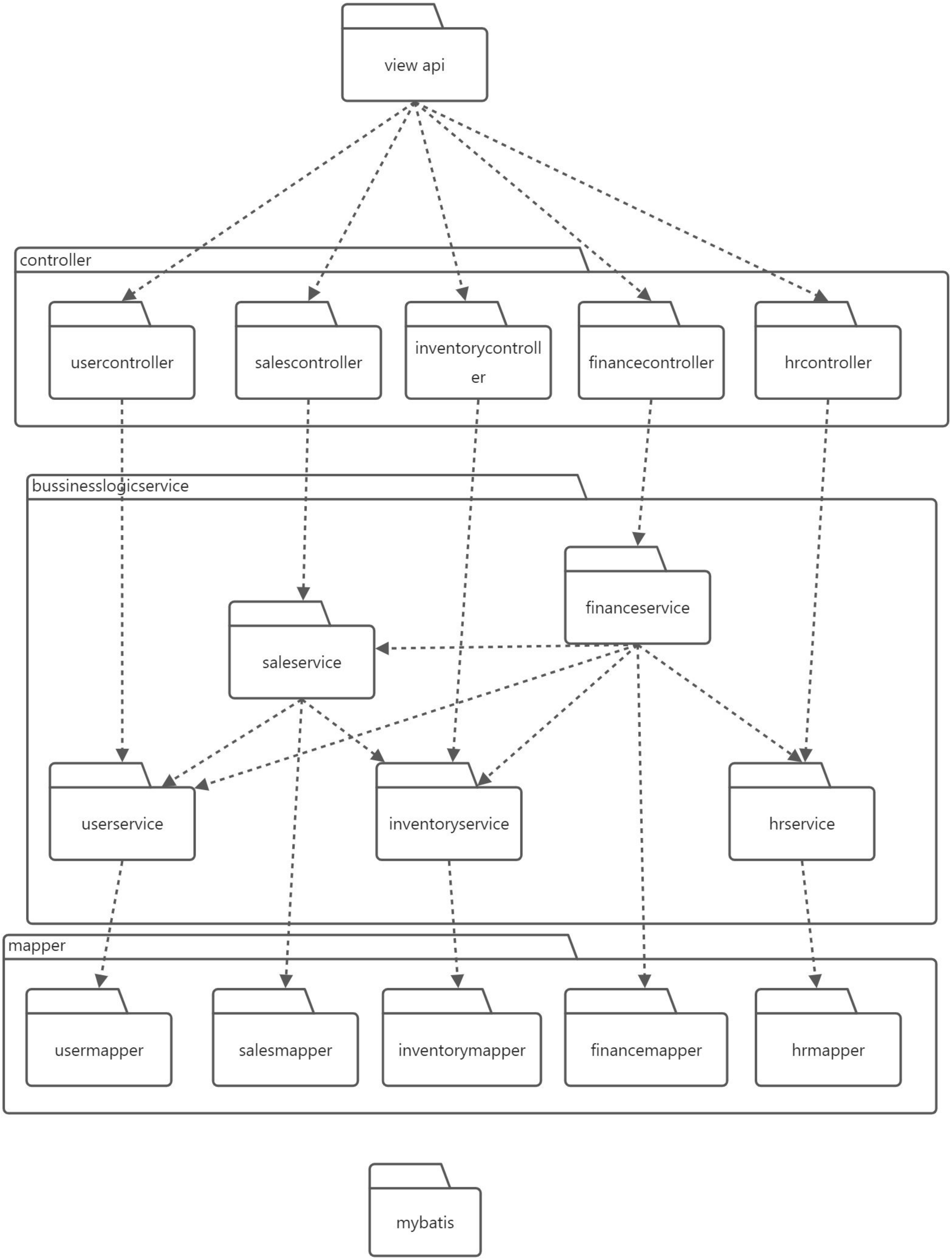


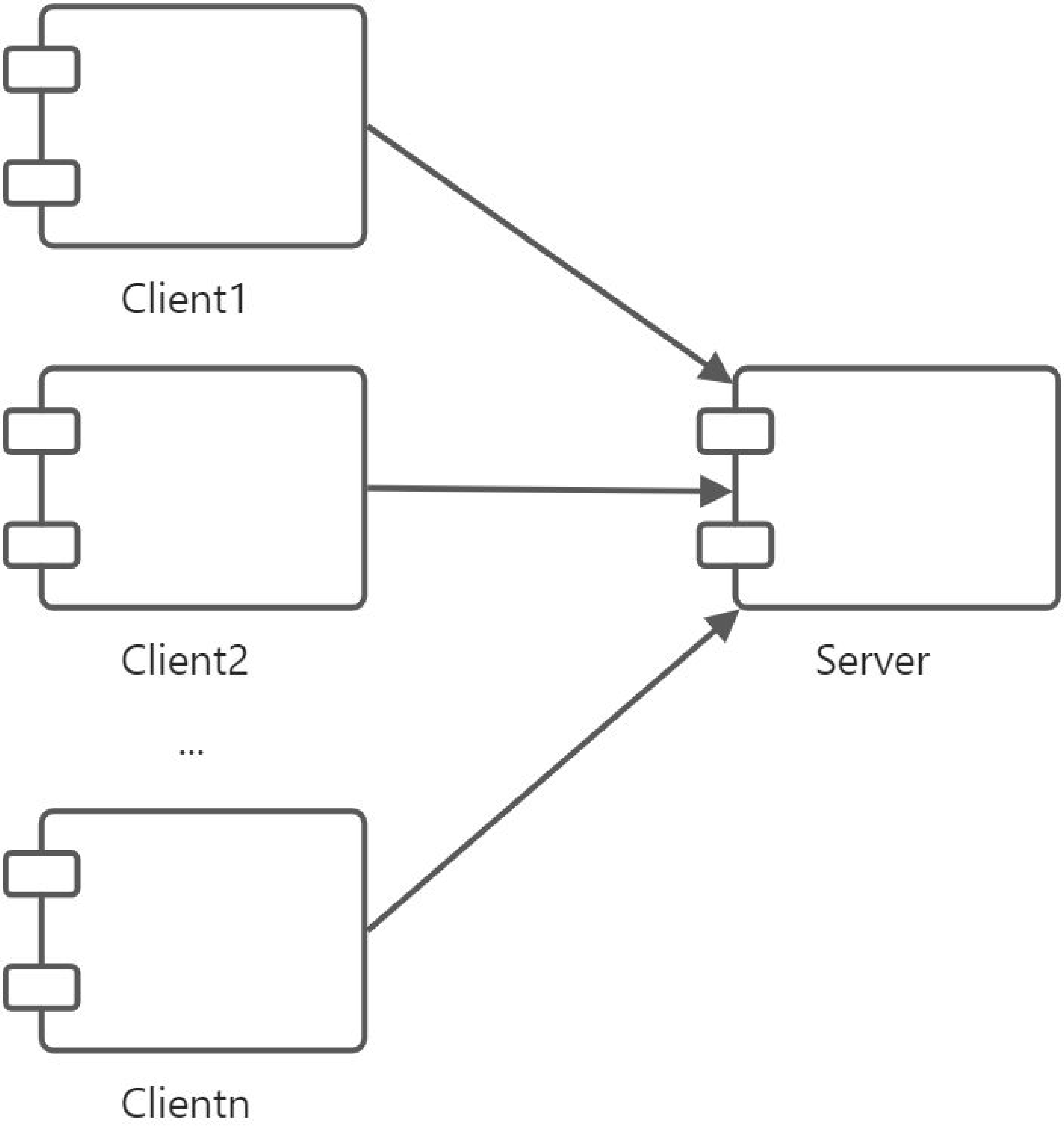
图4 ERP系统服务器端开发包图


view api
controller
usercontroller
salescontroller
inventorycontroll
er
financecontroller
hrcontroller
bussinesslogicservice
financeservice
saleservice
userservice
inventoryservice
hrservice
mapper
usermapper
salesmapper
inventorymapper
financemapper
hrmapper
mybatis

4.2运行时进程

在ERP系统中，会有多个客户端进程和一个服务器端进程，其进程图如图5所示。结合部署图，客户端进程是在客户端机器上运行，服务器端进程是在服务器端机器上运行。

图5 进程图



4.3物理部署

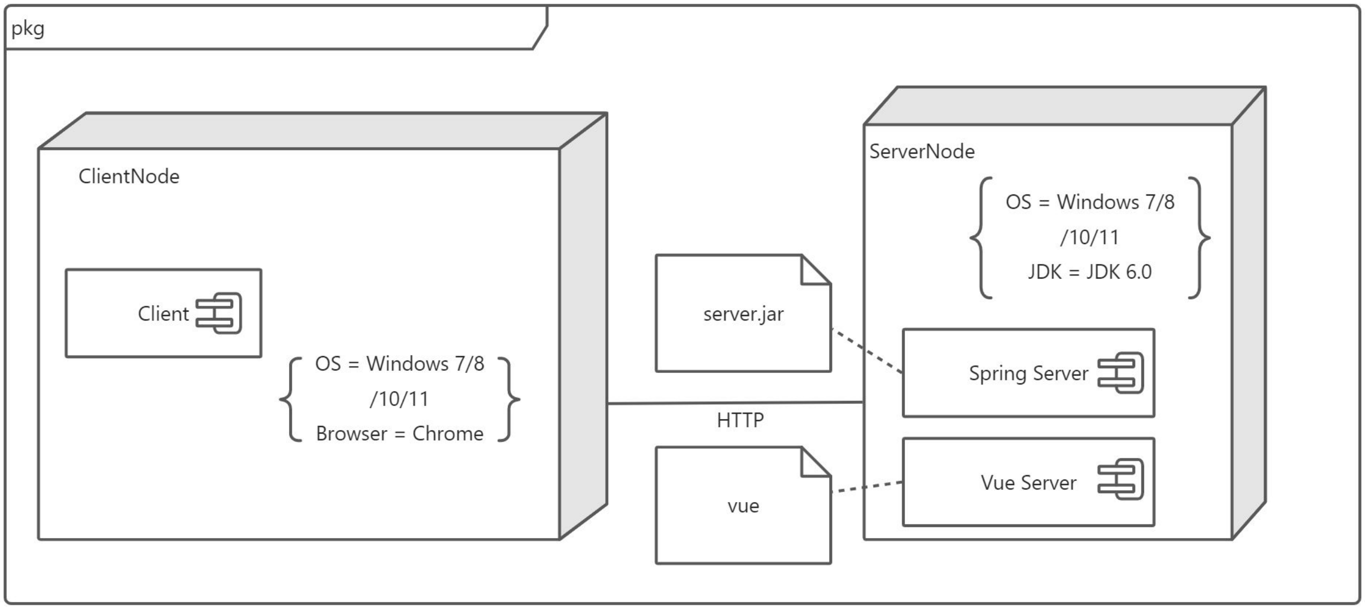


图6 部署图

5.接口视角

5.1模块的职责

图7 客户端模块视图

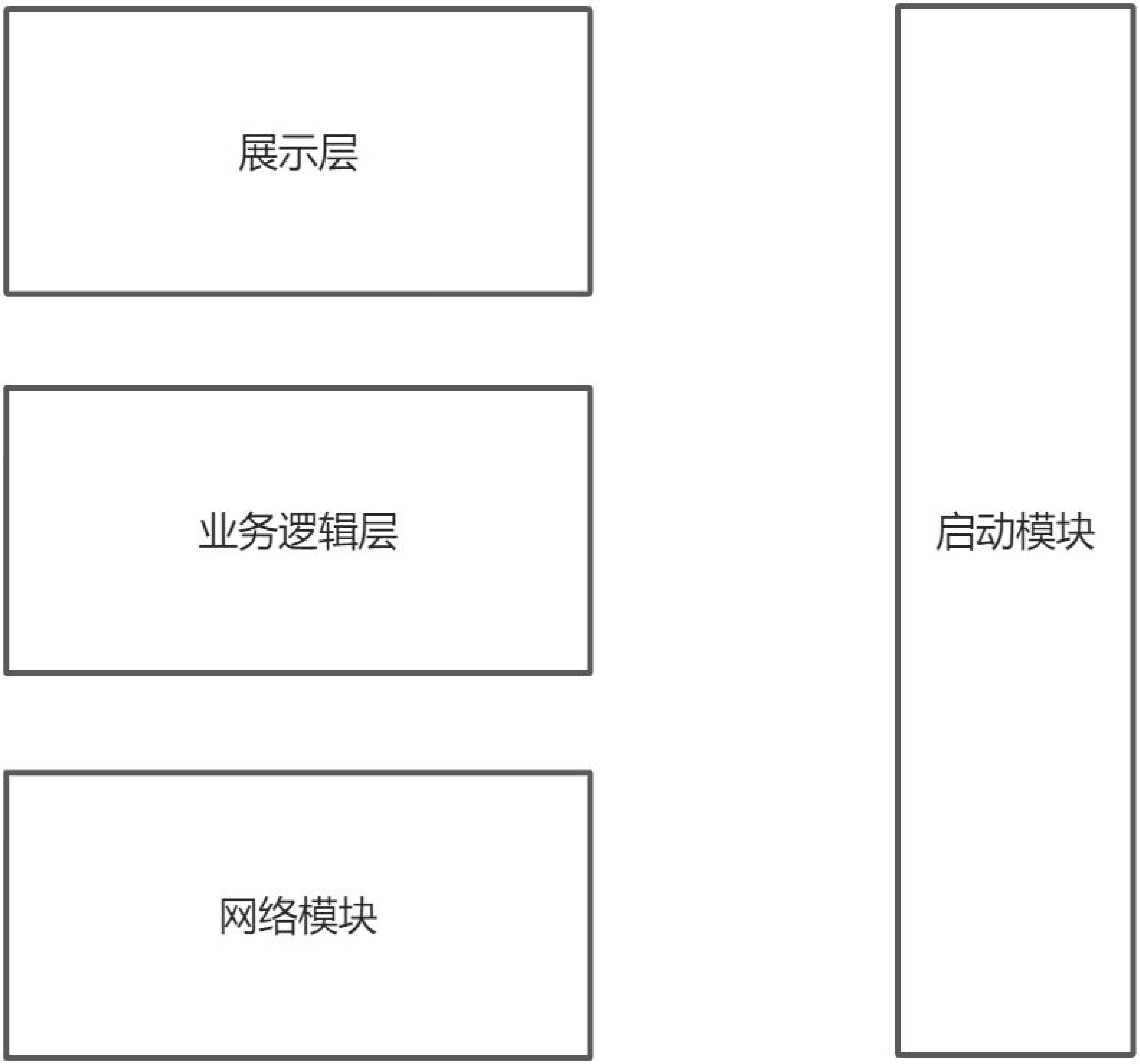


图8 服务器端模块视图

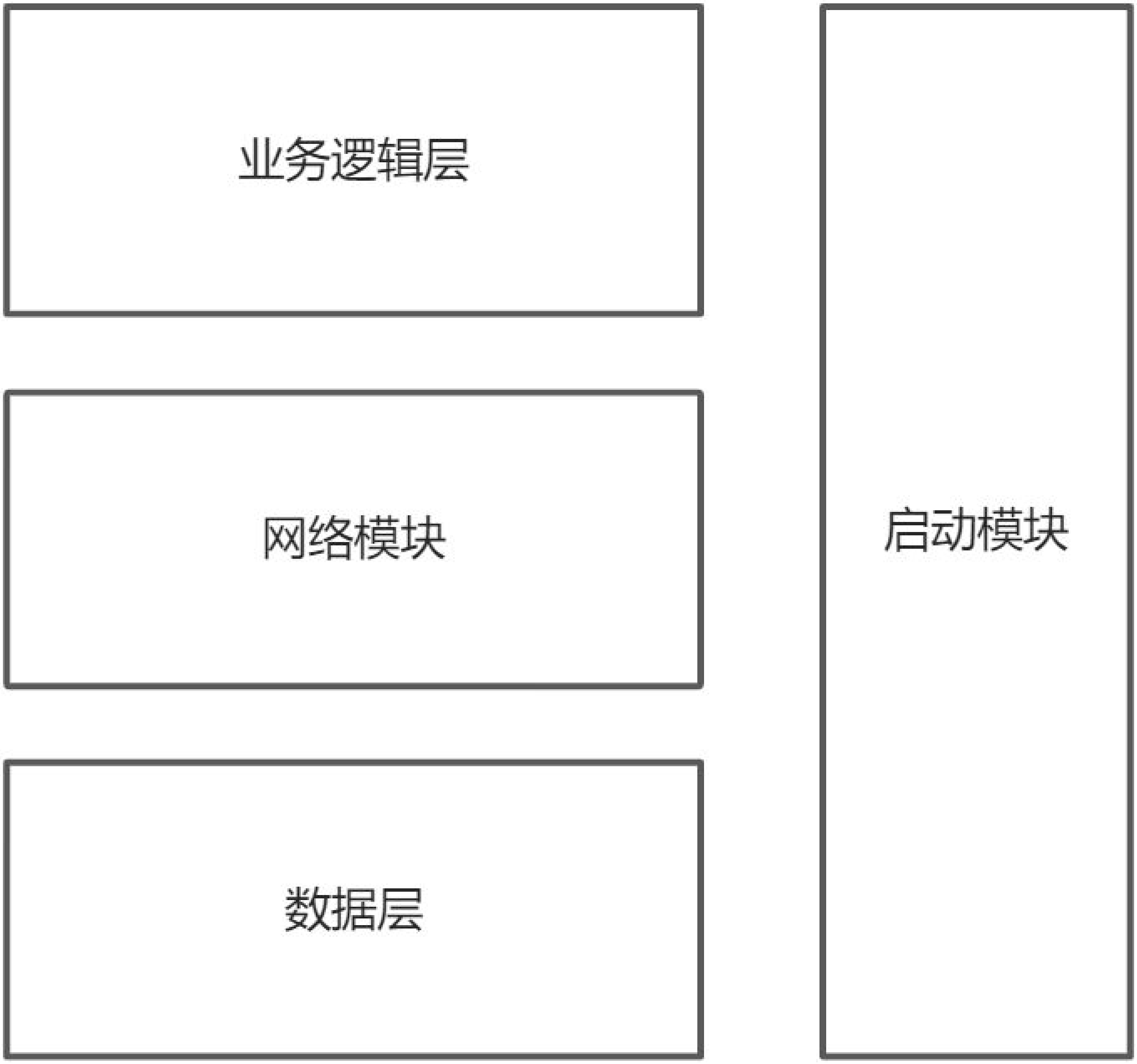


表2 客户端各层的职责

层	职责
启动模块	负责初始化网络通信机制，启动用户界面
用户界面层	基于vue的ERP客户端
客户端网络模块	利用http与服务器端通信

表3 服务器端各层的职责

层	职责
启动模块	负责初始化网络通信机制，启动用户界面
业务逻辑层	对客户端传入的的指令运行业务处理逻辑并传回结果数据
服务器端网络模块	利用http与服务器端通信
数据层	负责数据的持久化及数据访问接口

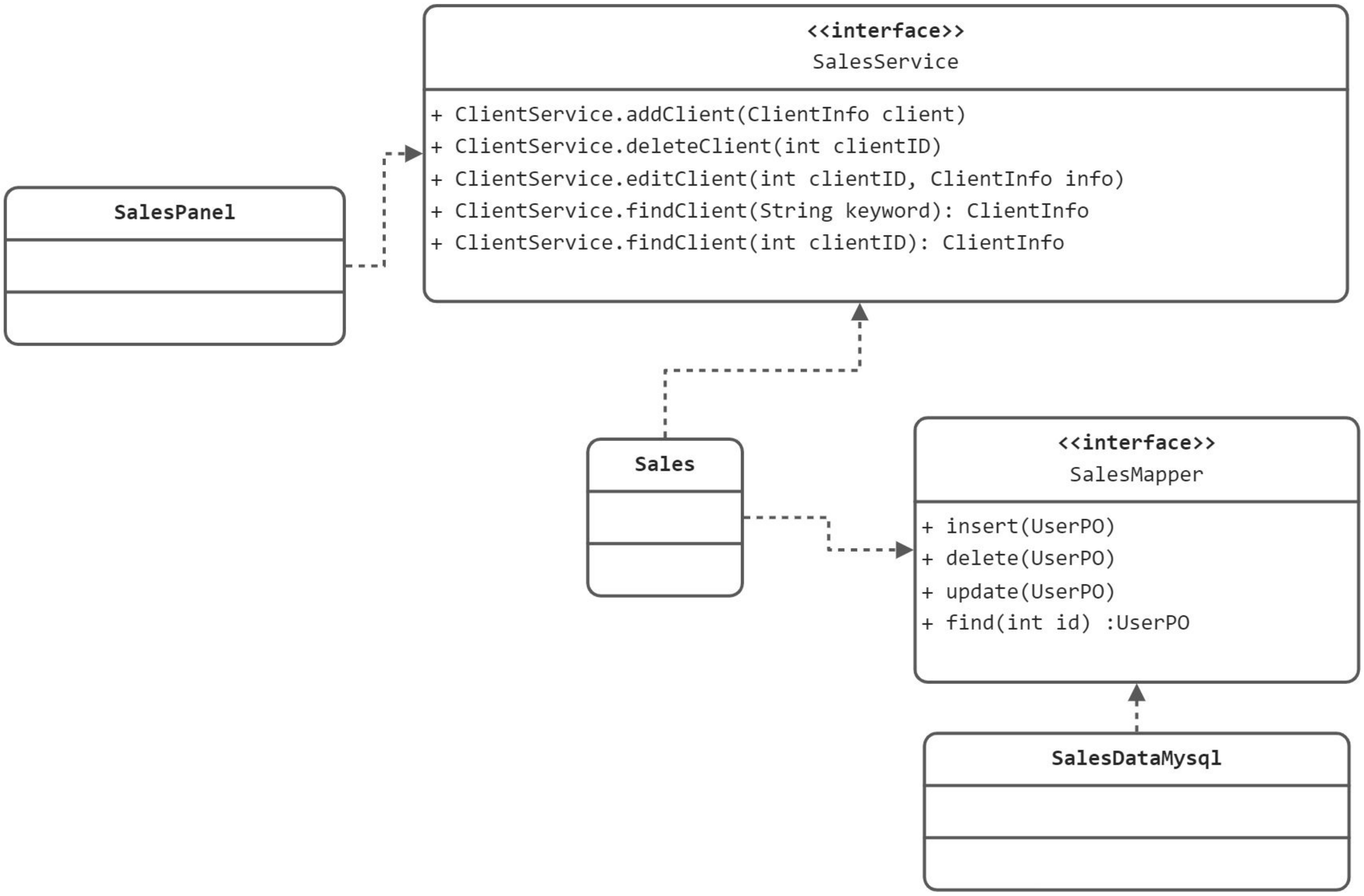
每一层只是使用下方直接接触的层。层与层之间仅仅是通过接口的调用来完成的。层之间调用的接口如表4所示。

表4 层之间调用的接口

接口	服务调用方	服务提供方
UserService SalesService InventoryService FinanceService HrService	客户端展示层	服务器端业务逻辑层
UserMapper SalesMapper InventoryMapper FinanceMapper HrMapper	服务器端业务逻辑层	服务器端数据层

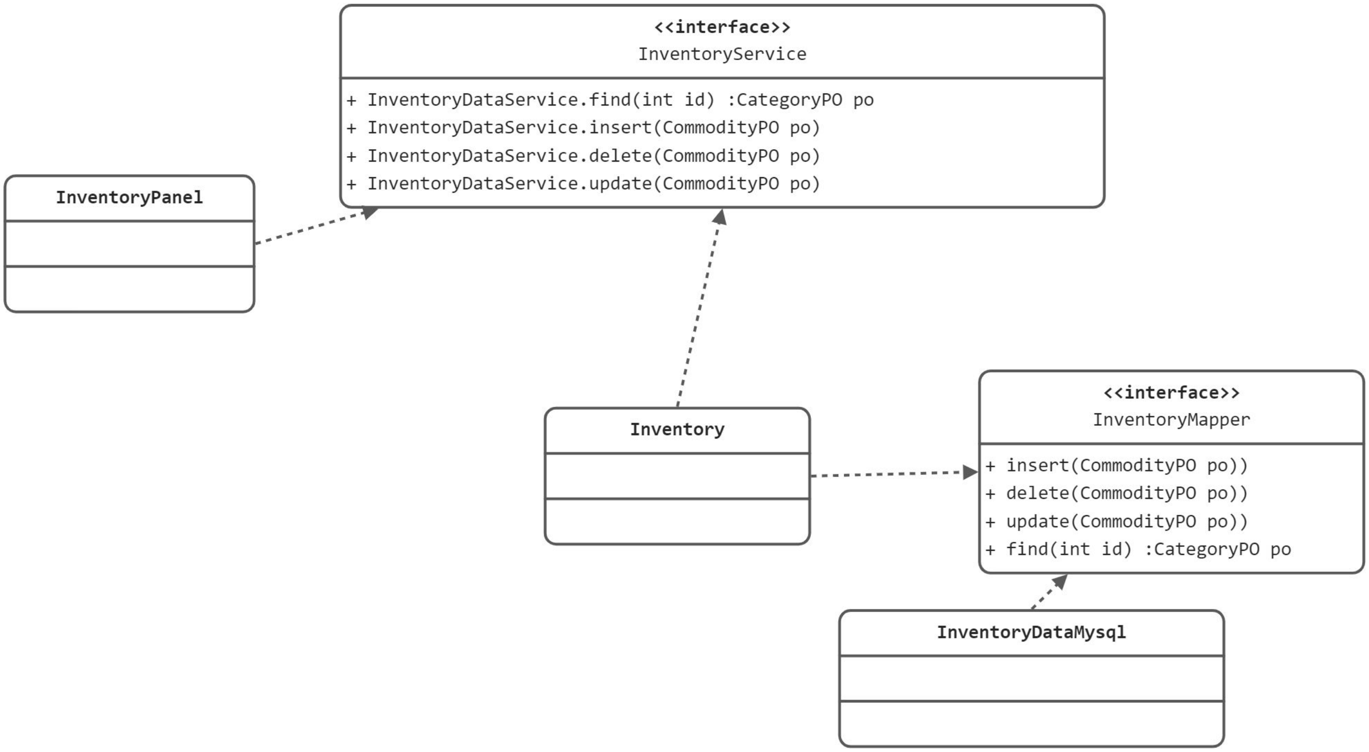
借用销售用例中的客户模块、库存商品部分用例来说明层之间的调用。

图9 销售用例层之间调用的接口



<<interface>>
SalesService
+ ClientService.addClient(ClientInfo client)
+ ClientService.deleteClient(int clientID)
+ ClientService.editClient(int clientID, ClientInfo info)
+ ClientService.findClient(String keyword): ClientInfo
+ ClientService.findClient(int clientID): ClientInfo
SalesPanel
Sales
<<interface>>
SalesMapper
+ insert(UserPO)
+ delete(UserPO)
+ update(UserPO)
+ find(int id) :UserPO
SalesDataMysql

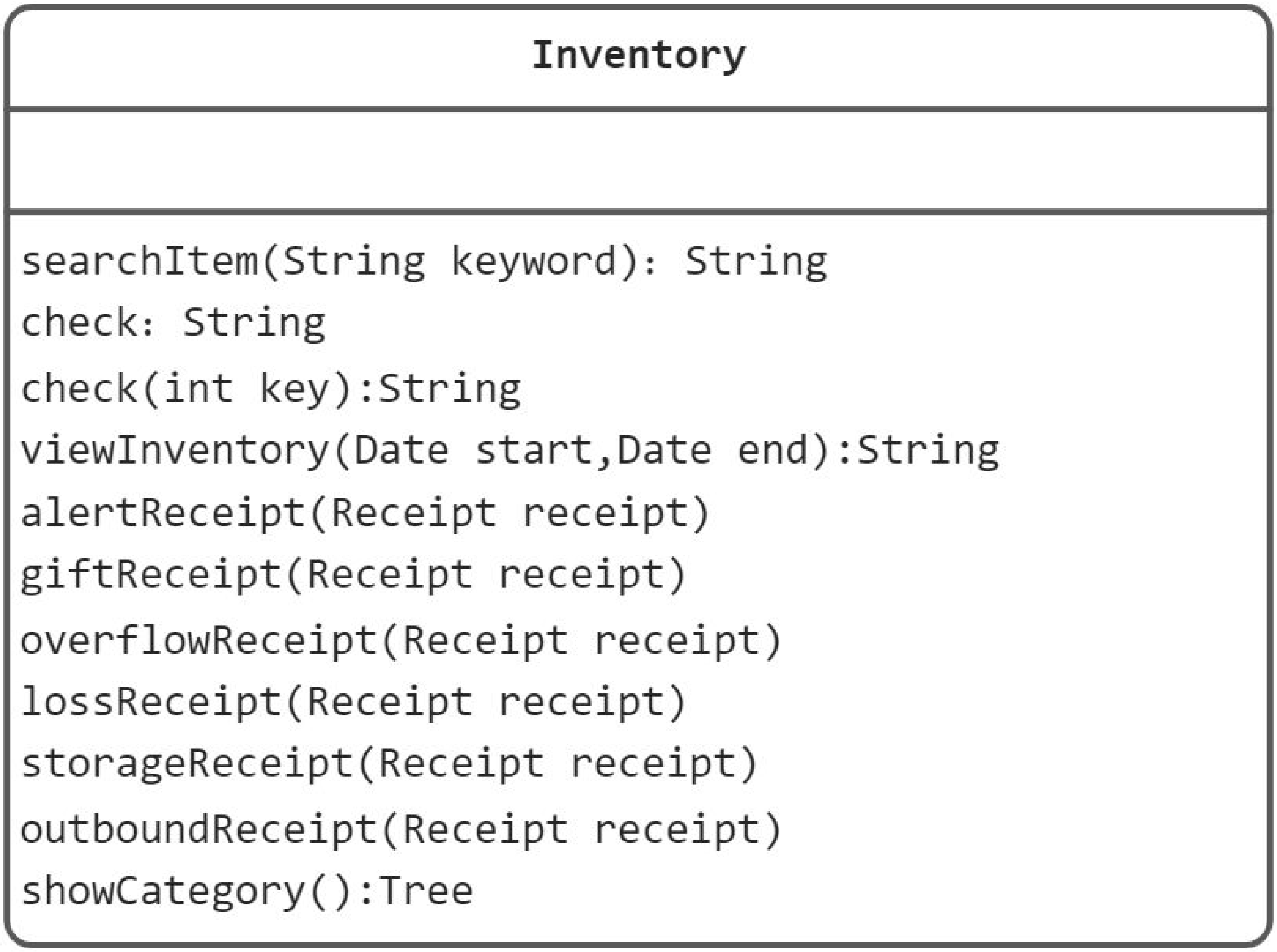
图10 库存用例之间调用的接口



<<interface>>
InventoryService
+ InventoryDataService.find(int id) :CategoryPO po
+ InventoryDataService.insert(CommodityPO po)
+ InventoryDataService.delete(CommodityPO po)
+ InventoryDataService.update(CommodityPO po)
InventoryPanel
Inventory
<<interface>>
InventoryMapper
+ insert(CommodityPO po))
+ delete(CommodityPO po))
+ update(CommodityPO po))
+ find(int id) :CategoryPO po
InventoryDataMysql

5.2库存模块的分解

图11 库存模块类


Inventory
searchItem(String keyword): String
check: String
check(int key):String
viewInventory(Date start,Date end):String
alertReceipt(Receipt receipt)
giftReceipt(Receipt receipt)
overflowReceipt(Receipt receipt)
lossReceipt(Receipt receipt)
storageReceipt(Receipt receipt)
outboundReceipt(Receipt receipt)
showCategory():Tree

5.2.1库存模块的职责

表5 库存模块的职责

模块	职责
InventoryController	负责库存界面的所需要的服务

5.2.2库存模块的接口规范

表6 库存模块的提供的接口规范

Inventory模块

提供的服务（供接口）

接口	语法	前置条件	后置条件
InventoryController.searchItem	selectItem(String keyword)	非空	展示商品搜索结果
InventoryController.searchItem	selectItem(int key)	非空	展示商品搜索结果
InventoryController.check	check()	非空	展示当天的各种商品的名称，型号，库存数量， 库存均价（商品的平均进价），批次批号，生产日期，并且显示行号。 Excel
InventoryController.viewInventory	viewInventory(Date start,Date end)	非空	查看此时间段内的出/入库数量/金额/商品信息/分类信息
InventoryController.alertReceipt	alertReceipt(Receipt receipt)	非空	向仓库管理员发送消息
InventoryController.giftReceipt	giftReceipt(Receipt receipt)	非空	库存信息进行更新
InventoryController.overflowReceipt	overflowReceipt(Receipt receipt)	非空	库存信息进行更新，保持一致
InventoryController.lossReceipt	lossReceipt(Receipt receipt)	非空	库存信息进行更新，保持一致
InventoryController.storageReceipt	storageReceipt(Receipt receipt)	非空	完成库存中信息的更改，并向库存管理人员发送信息
InventoryController.outboundReceipt	outboundReceipt(Receipt receipt)	非空	完成库存中信息的更改，并向库存管理人员发送信息
InventoryController.showCategory	showCategory()	无	显示库存中商品的分类，以树状

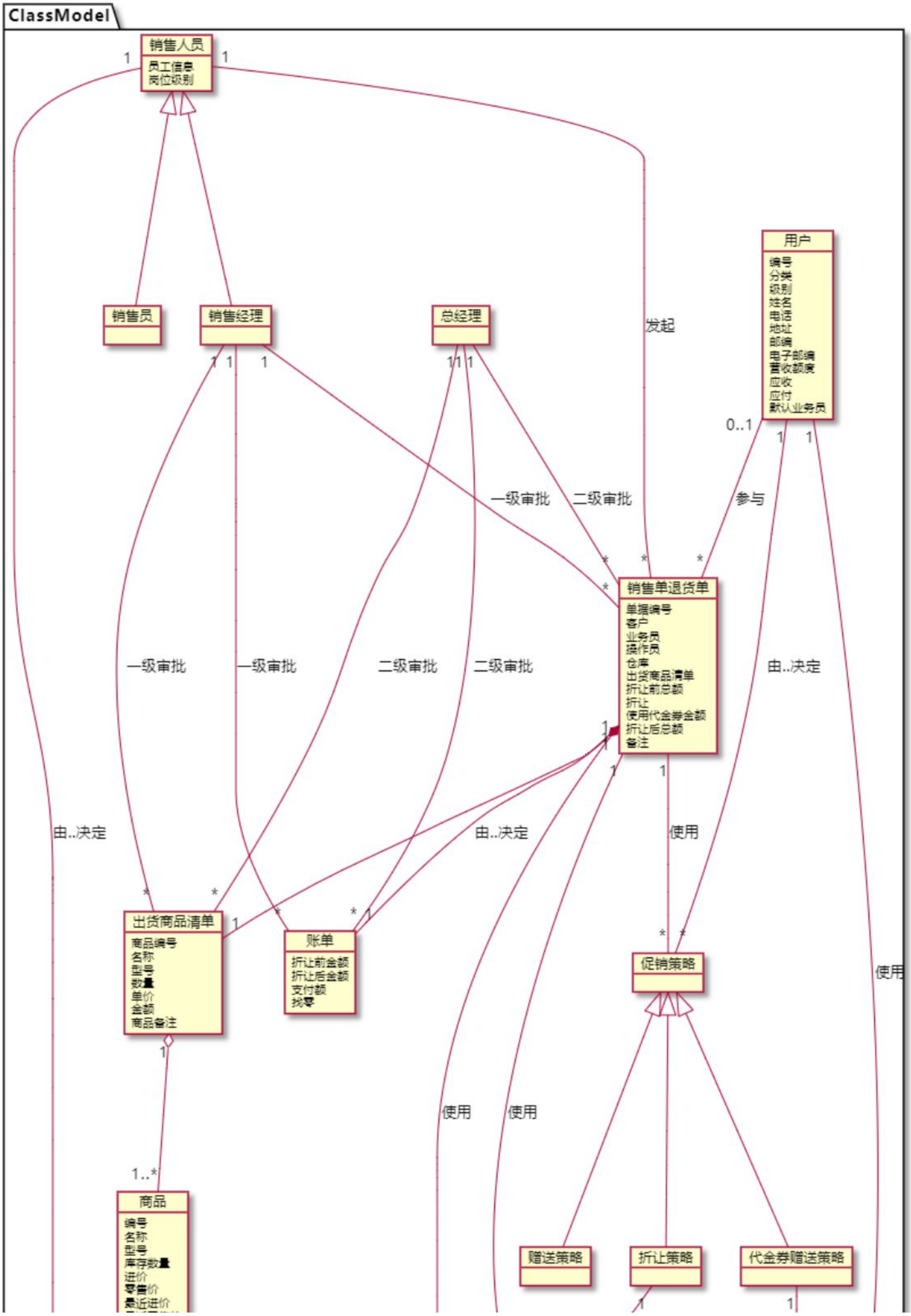
表7 库存模块的需要的服务接口

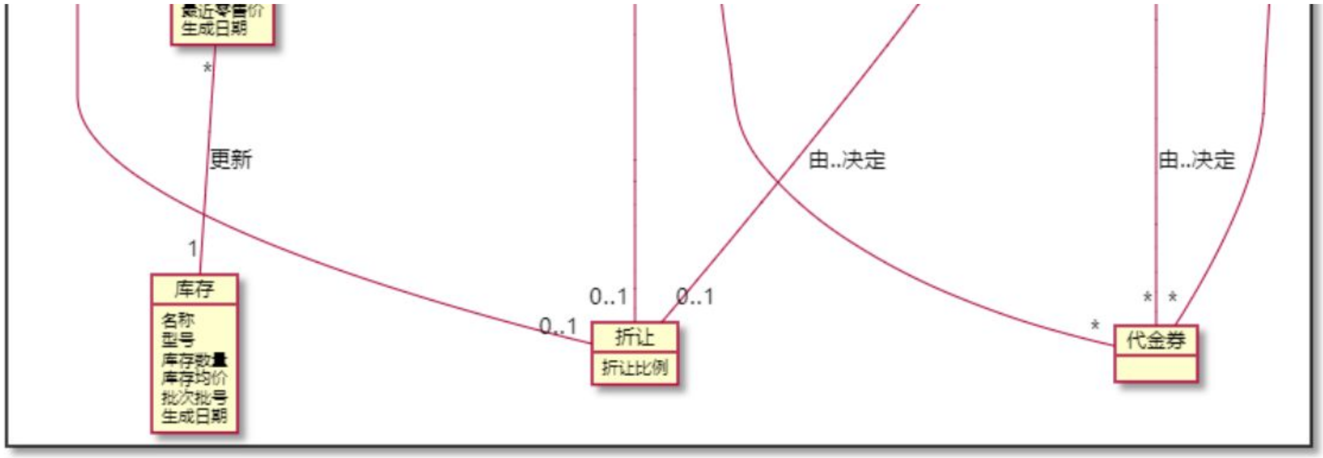
需要的服务

服务名	服务
InventoryDataService.find(int id)	根据编号进行查找单一持久化对象
InventoryDataService.insert(CommodityPO po)	插入单一持久化对象
InventoryDataService.delete(CommodityPO po)	删除单一持久化对象
InventoryDataService.update(CommmodityPO po)	更新单一持久化对象
DatabaseFactory.getDatabase	得到数据库的服务的引用
InventoryDataService.insertCategory(CategoryPO po)	插入单一持久化对象
InventoryDataService.deleteCategory(CategoryPO po)	删除单一持久化对象
InventoryDataService.updateCategory(CategoryPO po)	更新单一持久化对象

5.3销售模块的分解

图12 销售模块类


ClassModel
销售人员
员工信息
岗位级别
销售员
销售经理
总经理
用户
编号
分类
级别
姓名
电话
地址
邮编
电子邮编
营收额度
应收
应付
默认业务员
发起
参与
0..1
一级审批
二级审批
销售单退货单
单据编号
客户
业务员
操作员
仓库
出货商品清单
折让前总额
折让
使用代金券金额
折让后总额
备注
由..决定
使用
出货商品清单
商品编号
名称
型号
数量
单价
金额
商品备注
账单
折让前金额
折让后金额
支付额
找零
促销策略
1..*
商品
编号
名称
型号
库存数量
进价
零售价
最近进价
赠送策略
折让策略
代金券赠送策略
1
*



5.3.1销售模块的职责

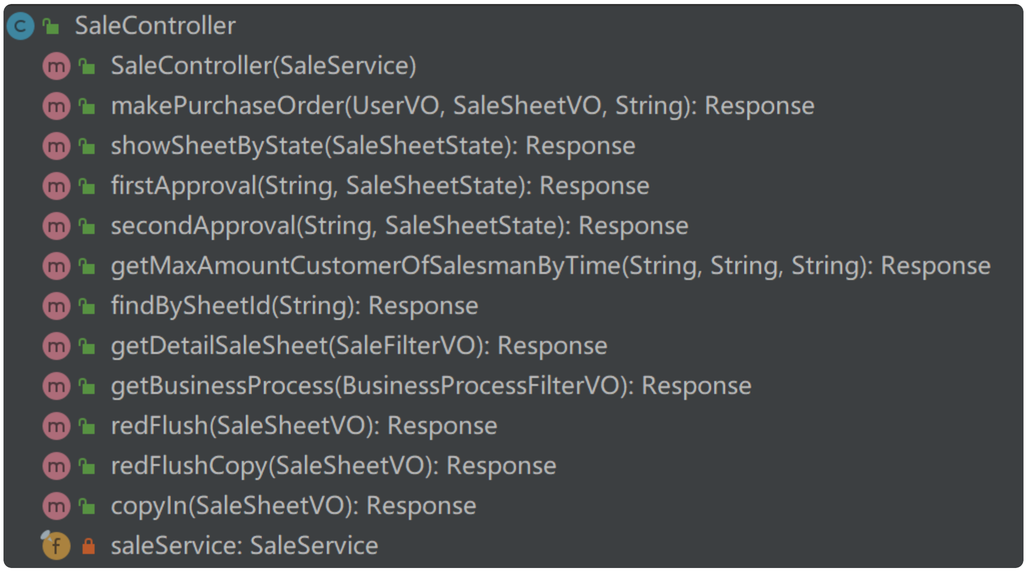
表8 销售模块的职责

模块	职责
SaleController	负责销售人员登陆相关
SaleRutrunController	负责折让相关
GiftSheetController	负责客户管理相关

销售模块接口规范

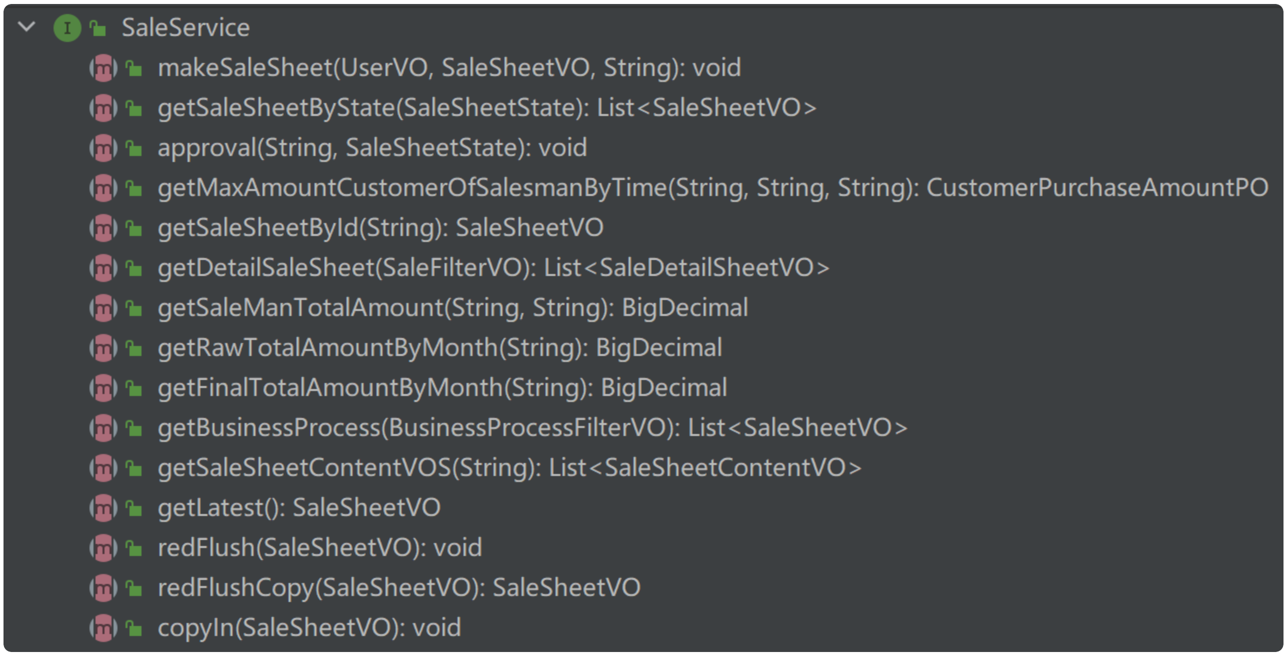
Sale模块

供接口



```
SaleController
    SaleController(SaleService)
    makePurchaseOrder(UserVO, SaleSheetVO, String): Response
    showSheetByState(SaleSheetState): Response
    firstApproval(String, SaleSheetState): Response
    secondApproval(String, SaleSheetState): Response
    getMaxAmountCustomerOfSalesmanByTime(String, String, String): Response
    findBySheetId(String): Response
    getDetailSaleSheet(SaleFilterVO): Response
    getBusinessProcess(BusinessProcessFilterVO): Response
    redFlush(SaleSheetVO): Response
    redFlushCopy(SaleSheetVO): Response
    copyIn(SaleSheetVO): Response
    saleService: SaleService
```

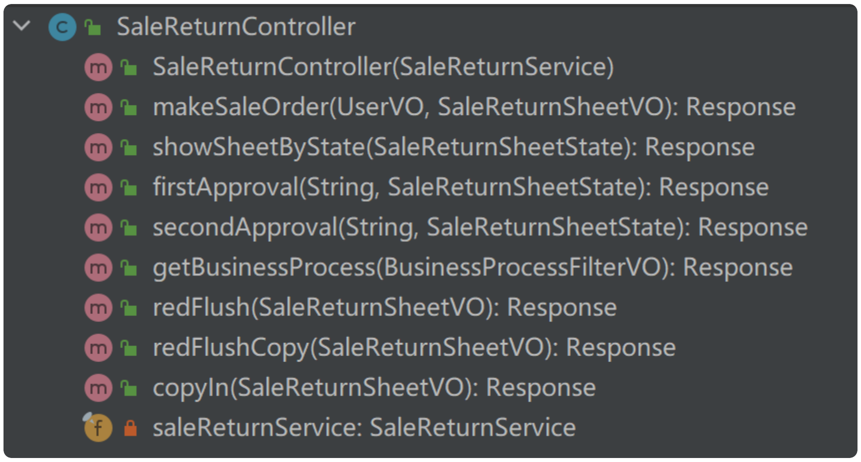
需接口



```
SaleService
    makeSaleSheet(UserVO, SaleSheetVO, String): void
    getSaleSheetByState(SaleSheetState): List<SaleSheetVO>
    approval(String, SaleSheetState): void
    getMaxAmountCustomerOfSalesmanByTime(String, String, String): CustomerPurchaseAmountPO
    getSaleSheetById(String): SaleSheetVO
    getDetailSaleSheet(SaleFilterVO): List<SaleDetailSheetVO>
    getSaleManTotalAmount(String, String): BigDecimal
    getRawTotalAmountByMonth(String): BigDecimal
    getFinalTotalAmountByMonth(String): BigDecimal
    getBusinessProcess(BusinessProcessFilterVO): List<SaleSheetVO>
    getSaleSheetContentVOS(String): List<SaleSheetContentVO>
    getLatest(): SaleSheetVO
    redFlush(SaleSheetVO): void
    redFlushCopy(SaleSheetVO): SaleSheetVO
    copyIn(SaleSheetVO): void
```

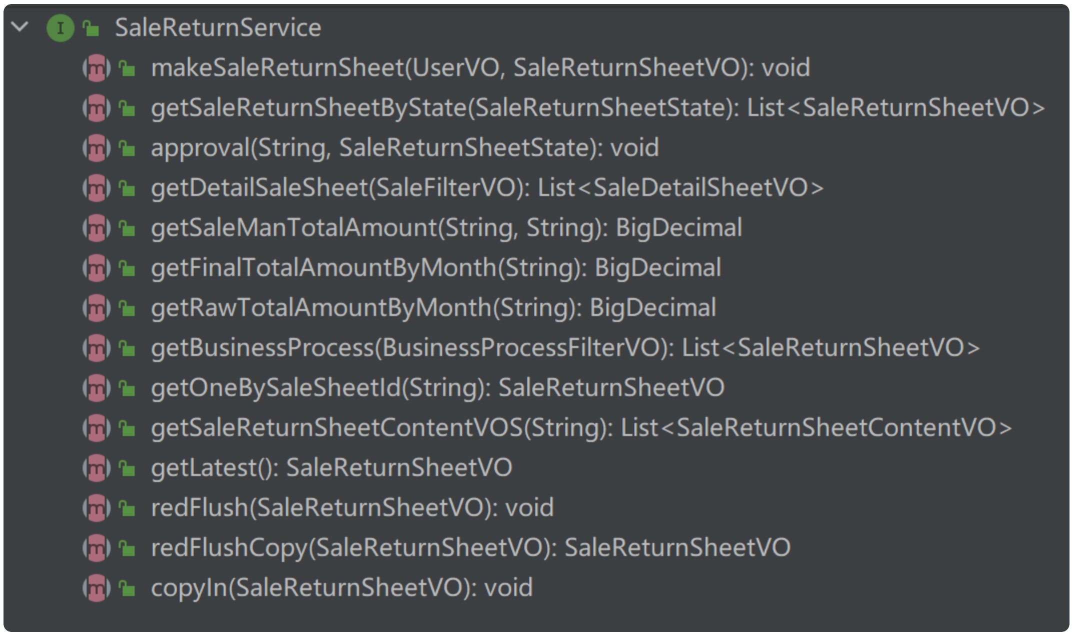
SaleRuturn模块

供接口



```
SaleReturnController
    SaleReturnController(SaleReturnService)
    makeSaleOrder(UserVO, SaleReturnSheetVO): Response
    showSheetByState(SaleReturnSheetState): Response
    firstApproval(String, SaleReturnSheetState): Response
    secondApproval(String, SaleReturnSheetState): Response
    getBusinessProcess(BusinessProcessFilterVO): Response
    redFlush(SaleReturnSheetVO): Response
    redFlushCopy(SaleReturnSheetVO): Response
    copyIn(SaleReturnSheetVO): Response
    saleReturnService: SaleReturnService
```

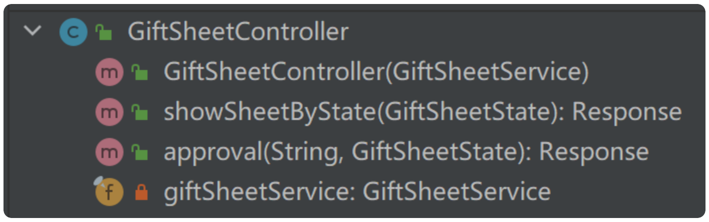
需接口



```
SaleReturnService
    makeSaleReturnSheet(UserVO, SaleReturnSheetVO): void
    getSaleReturnSheetByState(SaleReturnSheetState): List<SaleReturnSheetVO>
    approval(String, SaleReturnSheetState): void
    getDetailSaleSheet(SaleFilterVO): List<SaleDetailSheetVO>
    getSaleManTotalAmount(String, String): BigDecimal
    getFinalTotalAmountByMonth(String): BigDecimal
    getRawTotalAmountByMonth(String): BigDecimal
    getBusinessProcess(BusinessProcessFilterVO): List<SaleReturnSheetVO>
    getOneBySaleSheetId(String): SaleReturnSheetVO
    getSaleReturnSheetContentVOS(String): List<SaleReturnSheetContentVO>
    getLatest(): SaleReturnSheetVO
    redFlush(SaleReturnSheetVO): void
    redFlushCopy(SaleReturnSheetVO): SaleReturnSheetVO
    copyIn(SaleReturnSheetVO): void
```

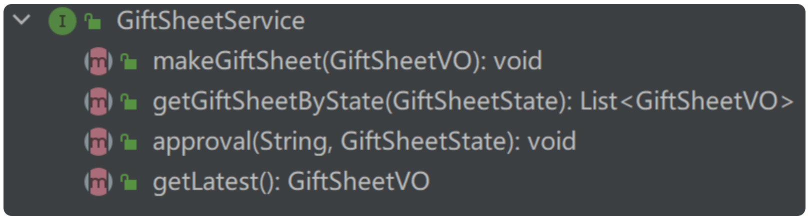
GiftSheet模块

供接口



```
GiftSheetController
    GiftSheetController(GiftSheetService)
    showSheetByState(GiftSheetState): Response
    approval(String, GiftSheetState): Response
    giftSheetService: GiftSheetService
```

需接口



```
GiftSheetService
    makeGiftSheet(GiftSheetVO): void
    getGiftSheetByState(GiftSheetState): List<GiftSheetVO>
    approval(String, GiftSheetState): void
    getLatest(): GiftSheetVO
```

5.4员工管理模块

5.4.1员工模块的职责

模块	职责
EmployeeController	负责员工的注册和增删改查相关
UserController	负责系统登陆，同时员工注册时会自动创建新账号

5.4.2员工模块接口规范

Employee模块

供接口

接口	语法	前置条件	后置条件
EmployeeController.makeEmployeeOrder	public Response makeEmployeeOrder(@RequestBody EmployeeVO employeeVO)	输入合法	将员工信息发送给EmployeeService
EmployeeController.showAll	public Response showAll()	输入合法	
EmployeeController.createEmployee	public Response createEmployee(@RequestBody EmployeeVO employeeVO)	输入合法	将员工id发送给EmployeeService
EmployeeController.deleteById	public Response deleteById(@RequestParam(value = "id") Integer employeeId)	输入合法	将员工信息发送给EmployeeService
EmployeeController.updateEmployee	public Response updateEmployee(@RequestBody EmployeeVO employeeVO)	输入合法	将员工信息发送给EmployeeService

需接口

服务名	服务
EmployeeVO create(EmployeeVO employeeVO);	新增员工信息
List<EmployeeVO> getAll();	获取员工列表
EmployeeVO getOneById(Integer employeeId);	获取指定员工
void deleteById(Integer employeeId);	删除指定员工
EmployeeVO update(EmployeeVO employeeVO);	更新员工
void register(UserVO userVO);	为员工自动注册系统账户

5.5折让管理模块

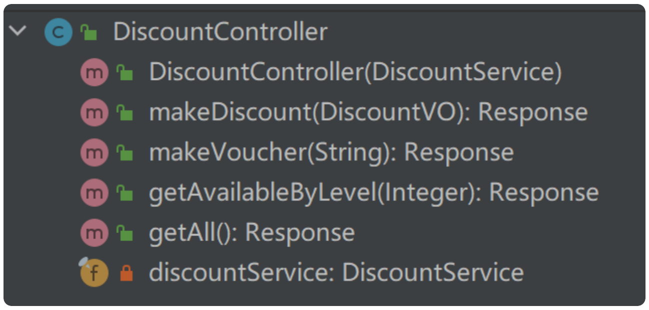
5.5.1折让管理模块职责

模块	职责
VoucherController	负责代金券，商品组合包、商品赠送相关
DiscountController	负责折扣相关

5.5.2折让模块接口规范

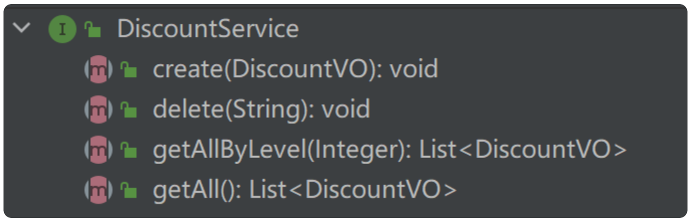
discount模块

供接口



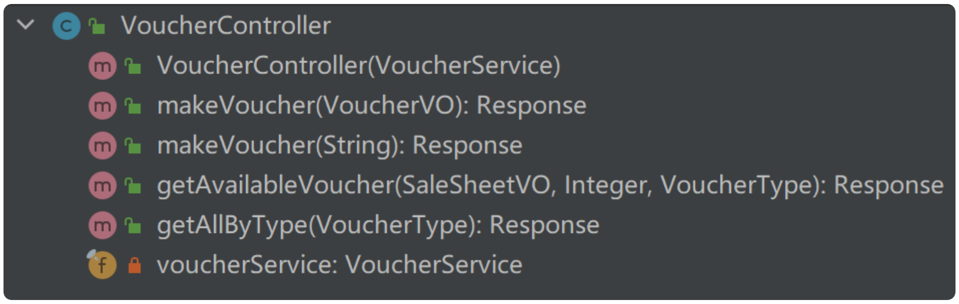
```
DiscountController
    DiscountController(DiscountService)
    makeDiscount(DiscountVO): Response
    makeVoucher(String): Response
    getAvailableByLevel(Integer): Response
    getAll(): Response
    discountService: DiscountService
```

需接口



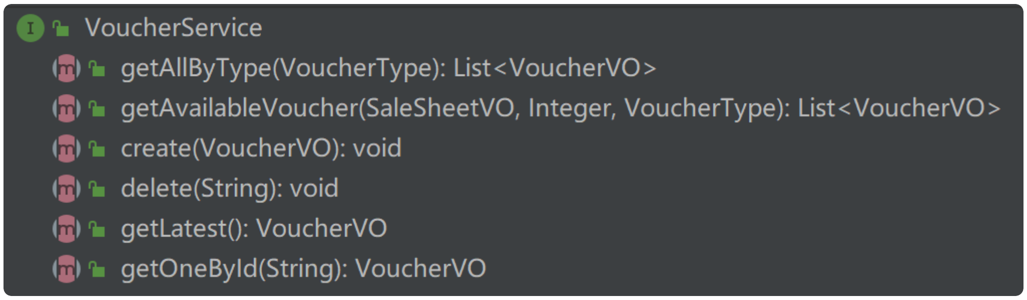
```
DiscountService
    create(DiscountVO): void
    delete(String): void
    getAllByLevel(Integer): List<DiscountVO>
    getAll(): List<DiscountVO>
```

Voucher模块



```
VoucherController
    VoucherController(VoucherService)
    makeVoucher(VoucherVO): Response
    makeVoucher(String): Response
    getAvailableVoucher(SaleSheetVO, Integer, VoucherType): Response
    getAllByType(VoucherType): Response
    voucherService: VoucherService
```

需接口



```
VoucherService
    getAllByType(VoucherType): List<VoucherVO>
    getAvailableVoucher(SaleSheetVO, Integer, VoucherType): List<VoucherVO>
    create(VoucherVO): void
    delete(String): void
    getLatest(): VoucherVO
    getOneById(String): VoucherVO
```

5.6工资税收管理模块

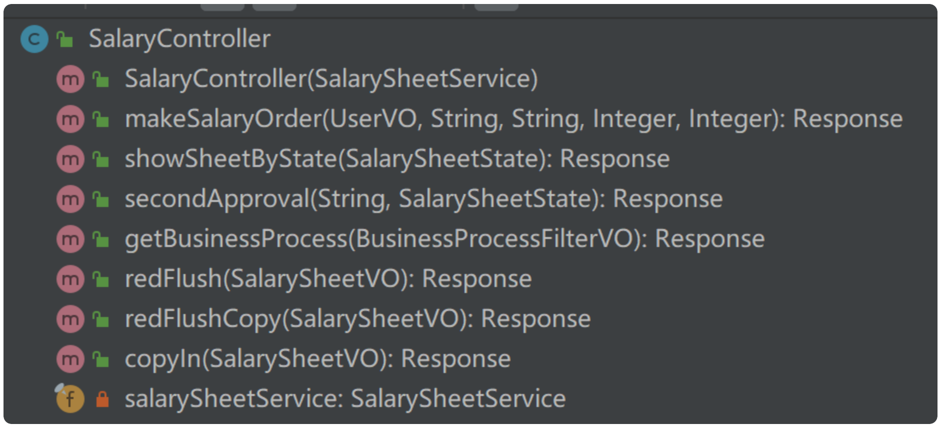
5.6.1工资税收管理模块职责

模块	职责
SalarySystemController	负责薪酬制度相关相关
SalaryController	负责工资相关
AnnualBonusController	负责年终奖相关

5.6.2工资税收管理模块接口

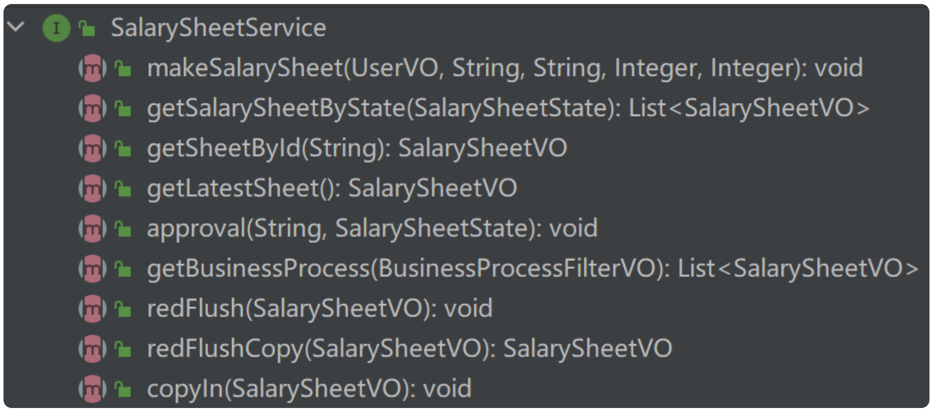
SalarySheet模块

供接口

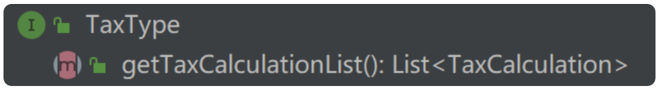


```
SalaryController
    SalaryController(SalarySheetService)
    makeSalaryOrder(UserVO, String, String, Integer, Integer): Response
    showSheetByState(SalarySheetState): Response
    secondApproval(String, SalarySheetState): Response
    getBusinessProcess(BusinessProcessFilterVO): Response
    redFlush(SalarySheetVO): Response
    redFlushCopy(SalarySheetVO): Response
    copyIn(SalarySheetVO): Response
    salarySheetService: SalarySheetService
```

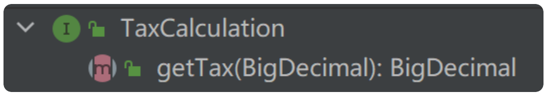
需接口



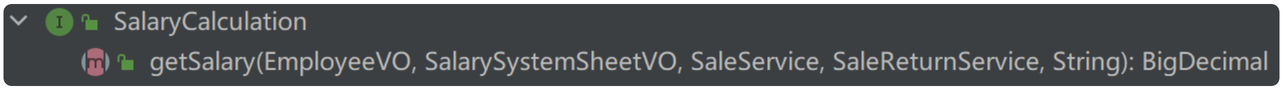
```
SalarySheetService
    makeSalarySheet(UserVO, String, String, Integer, Integer): void
    getSalarySheetByState(SalarySheetState): List<SalarySheetVO>
    getSheetById(String): SalarySheetVO
    getLatestSheet(): SalarySheetVO
    approval(String, SalarySheetState): void
    getBusinessProcess(BusinessProcessFilterVO): List<SalarySheetVO>
    redFlush(SalarySheetVO): void
    redFlushCopy(SalarySheetVO): SalarySheetVO
    copyIn(SalarySheetVO): void
```



```
TaxType
    getTaxCalculationList(): List<TaxCalculation>
```



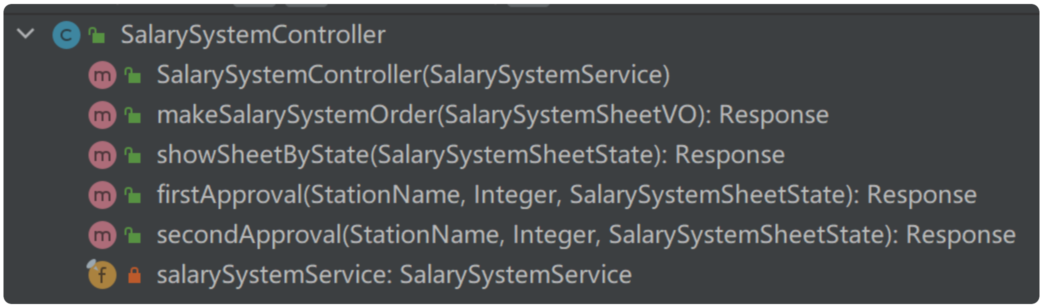
```
TaxCalculation
    getTax(BigDecimal): BigDecimal
```



```
SalaryCalculation
    getSalary(EmployeeVO, SalarySystemSheetVO, SaleService, SaleReturnService, String): BigDecimal
```

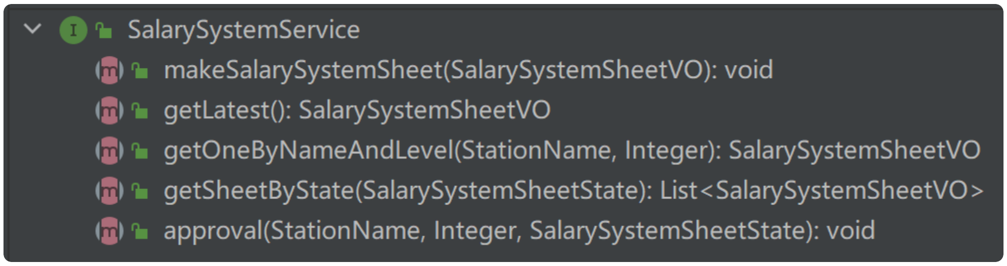
SalarySystem模块

供接口

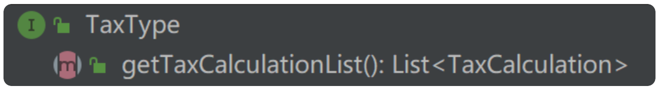


```
SalarySystemController
    SalarySystemController(SalarySystemService)
    makeSalarySystemOrder(SalarySystemSheetVO): Response
    showSheetByState(SalarySystemSheetState): Response
    firstApproval(StationName, Integer, SalarySystemSheetState): Response
    secondApproval(StationName, Integer, SalarySystemSheetState): Response
    salarySystemService: SalarySystemService
```

需接口



```
SalarySystemService
    makeSalarySystemSheet(SalarySystemSheetVO): void
    getLatest(): SalarySystemSheetVO
    getOneByNameAndLevel(StationName, Integer): SalarySystemSheetVO
    getSheetByState(SalarySystemSheetState): List<SalarySystemSheetVO>
    approval(StationName, Integer, SalarySystemSheetState): void
```



```
TaxType
    getTaxCalculationList(): List<TaxCalculation>
```

5.7账户转账模块

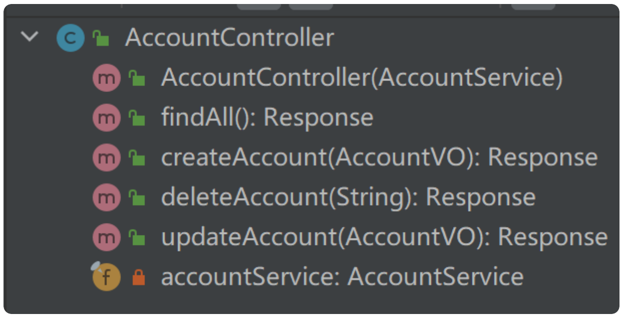
5.7.1账户转账模块职责

模块	职责
AccountController	负责公司账户相关
PayableController	负责付款单相关
ReceiveController	负责收款单相关
TransferController	负责转账提醒

5.7.2账户转账模块的接口

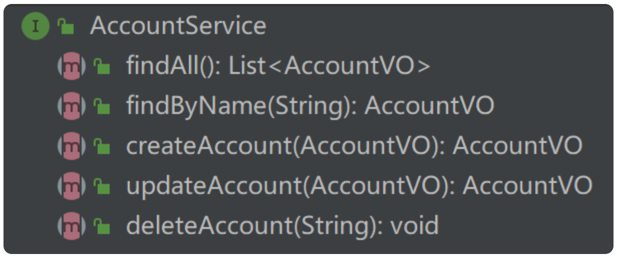
account模块

供接口



```
AccountController
    AccountController(AccountService)
    findAll(): Response
    createAccount(AccountVO): Response
    deleteAccount(String): Response
    updateAccount(AccountVO): Response
    accountService: AccountService
```

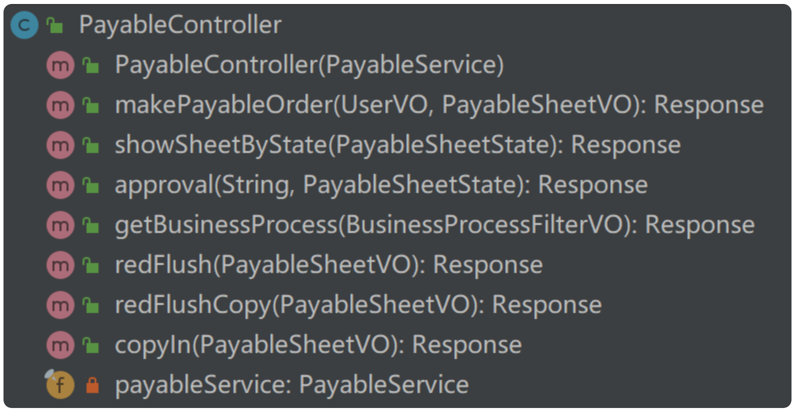
需接口



```
AccountService
    findAll(): List<AccountVO>
    findByName(String): AccountVO
    createAccount(AccountVO): AccountVO
    updateAccount(AccountVO): AccountVO
    deleteAccount(String): void
```

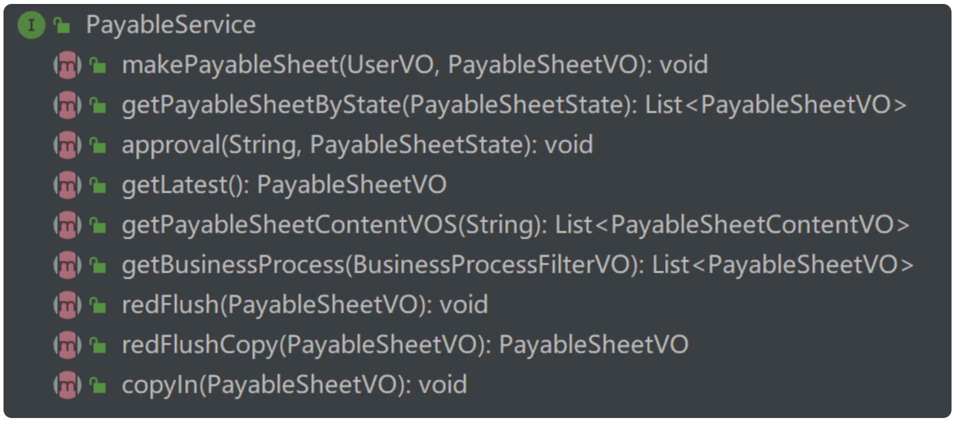
payable模块

供接口



```
PayableController
    PayableController(PayableService)
    makePayableOrder(UserVO, PayableSheetVO): Response
    showSheetByState(PayableSheetState): Response
    approval(String, PayableSheetState): Response
    getBusinessProcess(BusinessProcessFilterVO): Response
    redFlush(PayableSheetVO): Response
    redFlushCopy(PayableSheetVO): Response
    copyIn(PayableSheetVO): Response
    payableService: PayableService
```

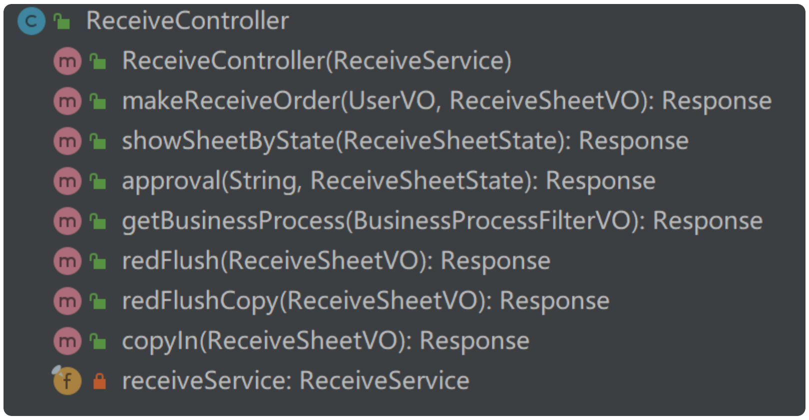
需接口



```
PayableService
    makePayableSheet(UserVO, PayableSheetVO): void
    getPayableSheetByState(PayableSheetState): List<PayableSheetVO>
    approval(String, PayableSheetState): void
    getLatest(): PayableSheetVO
    getPayableSheetContentVOS(String): List<PayableSheetContentVO>
    getBusinessProcess(BusinessProcessFilterVO): List<PayableSheetVO>
    redFlush(PayableSheetVO): void
    redFlushCopy(PayableSheetVO): PayableSheetVO
    copyIn(PayableSheetVO): void
```

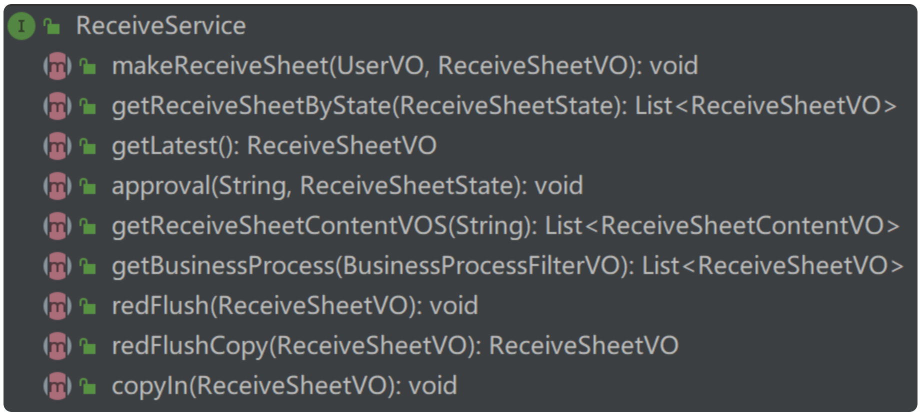
receive模块

供接口



```
ReceiveController
    ReceiveController(ReceiveService)
    makeReceiveOrder(UserVO, ReceiveSheetVO): Response
    showSheetByState(ReceiveSheetState): Response
    approval(String, ReceiveSheetState): Response
    getBusinessProcess(BusinessProcessFilterVO): Response
    redFlush(ReceiveSheetVO): Response
    redFlushCopy(ReceiveSheetVO): Response
    copyIn(ReceiveSheetVO): Response
    receiveService: ReceiveService
```

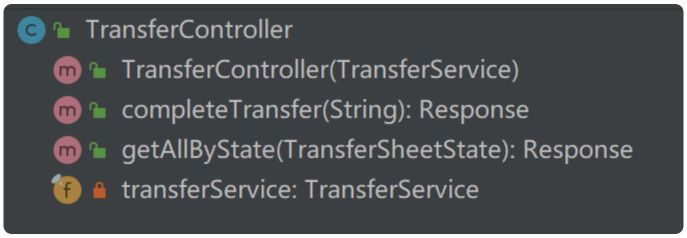
需接口



```
ReceiveService
    makeReceiveSheet(UserVO, ReceiveSheetVO): void
    getReceiveSheetByState(ReceiveSheetState): List<ReceiveSheetVO>
    getLatest(): ReceiveSheetVO
    approval(String, ReceiveSheetState): void
    getReceiveSheetContentVOS(String): List<ReceiveSheetContentVO>
    getBusinessProcess(BusinessProcessFilterVO): List<ReceiveSheetVO>
    redFlush(ReceiveSheetVO): void
    redFlushCopy(ReceiveSheetVO): ReceiveSheetVO
    copyIn(ReceiveSheetVO): void
```

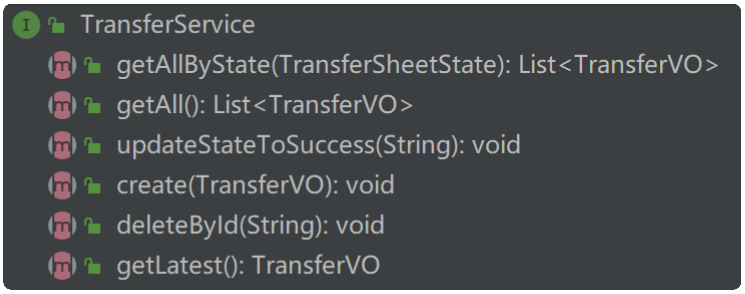
transfer模块

供接口



```
TransferController
    TransferController(TransferService)
    completeTransfer(String): Response
    getAllByState(TransferSheetState): Response
    transferService: TransferService
```

需接口



```
TransferService
    getAllByState(TransferSheetState): List<TransferVO>
    getAll(): List<TransferVO>
    updateStateToSuccess(String): void
    create(TransferVO): void
    deleteById(String): void
    getLatest(): TransferVO
```

5.8经营数据查看模块

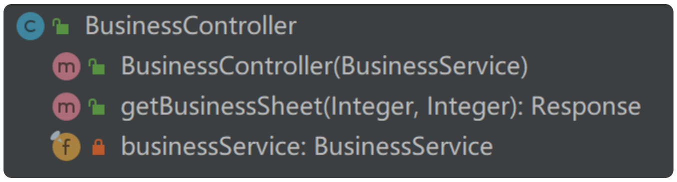
5.8.1经营情况查看模块职责

模块	职责
BusinessController	负责经营情况查看相关
SheetExportController	负责数据导出导入相关

5.8.2经营情况查看模块的接口

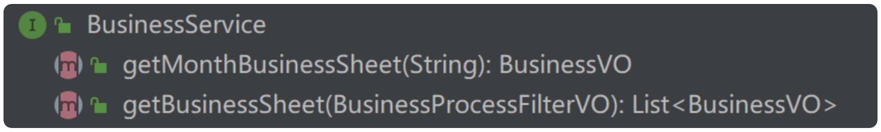
Business模块

供接口

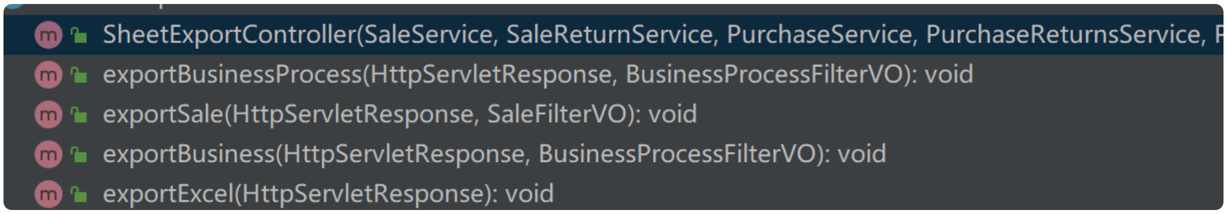


```
BusinessController
    BusinessController(BusinessService)
    getBusinessSheet(Integer, Integer): Response
    businessService: BusinessService
```

需接口



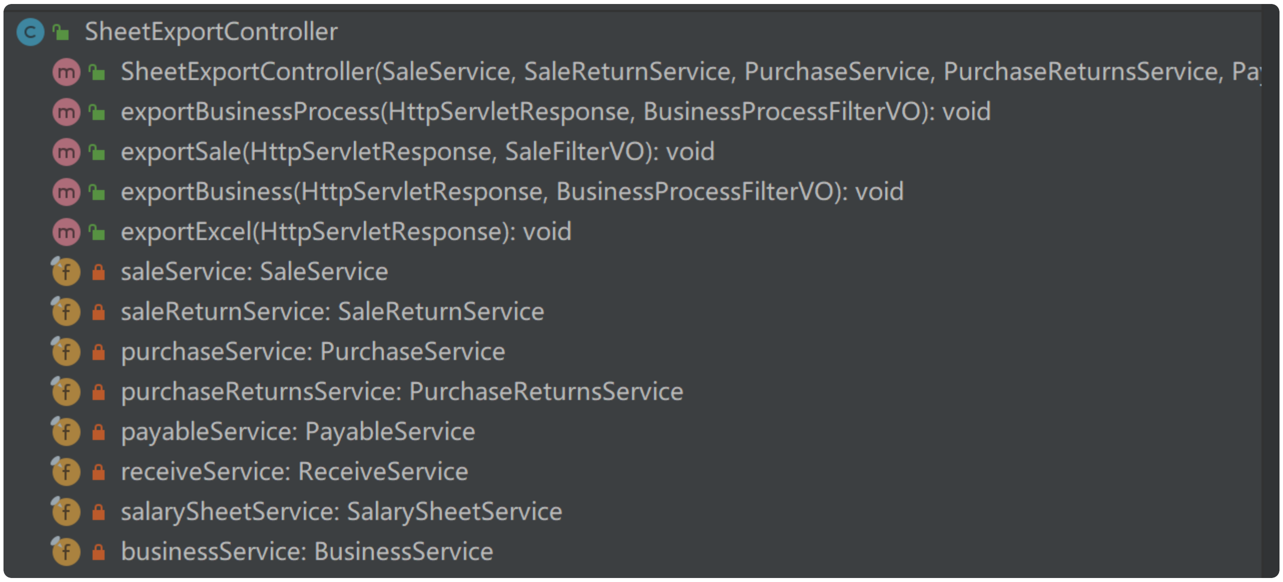
```
BusinessService
    getMonthBusinessSheet(String): BusinessVO
    getBusinessSheet(BusinessProcessFilterVO): List<BusinessVO>
```



```
    SheetExportController(SaleService, SaleReturnService, PurchaseService, PurchaseReturnsService, P
    exportBusinessProcess(HttpServletResponse, BusinessProcessFilterVO): void
    exportSale(HttpServletResponse, SaleFilterVO): void
    exportBusiness(HttpServletResponse, BusinessProcessFilterVO): void
    exportExcel(HttpServletResponse): void
```

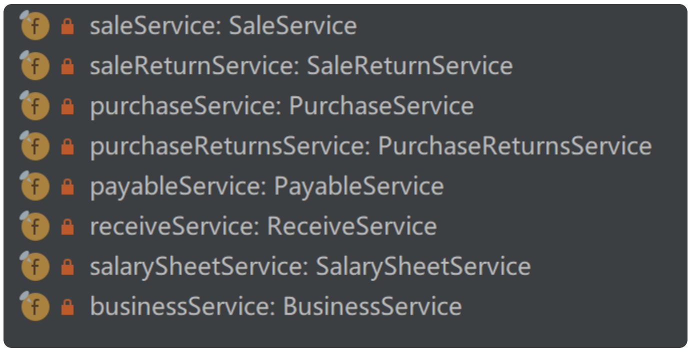
SheetExport模块

供接口



```
SheetExportController
    SheetExportController(SaleService, SaleReturnService, PurchaseService, PurchaseReturnsService, Pa
    exportBusinessProcess(HttpServletResponse, BusinessProcessFilterVO): void
    exportSale(HttpServletResponse, SaleFilterVO): void
    exportBusiness(HttpServletResponse, BusinessProcessFilterVO): void
    exportExcel(HttpServletResponse): void
    saleService: SaleService
    saleReturnService: SaleReturnService
    purchaseService: PurchaseService
    purchaseReturnsService: PurchaseReturnsService
    payableService: PayableService
    receiveService: ReceiveService
    salarySheetService: SalarySheetService
    businessService: BusinessService
```

需接口



```
saleService: SaleService
saleReturnService: SaleReturnService
purchaseService: PurchaseService
purchaseReturnsService: PurchaseReturnsService
payableService: PayableService
receiveService: ReceiveService
salarySheetService: SalarySheetService
businessService: BusinessService
```

5.9系统管理模块

5.9.1系统管理模块职责

管理用户登陆和系统备份（初期建账功能）

模块	职责
LoginController	负责销售人员登陆相关
BackupController	负责初期建账相关

Login模块

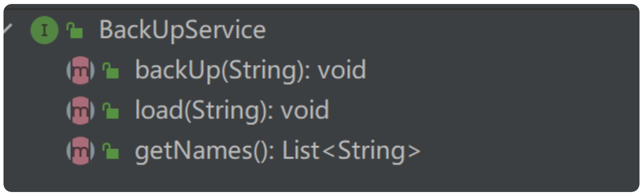
供接口

接口	语法	前置条件	后置条件
LoginController.getUserInput	getUserInput(String username, String pwd)	输入合法	从LoginService返回登陆状态信息
LoginController.showLoginState	showLoginState(LoginState state)	无	在页面显示登陆状态（成功、失败）

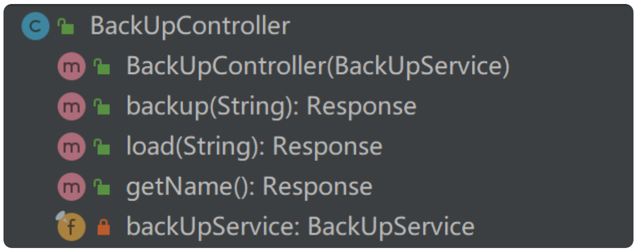
需接口

服务名	服务
LoginService.checkUserGroup	检查用户输入信息，若存在，检查用户组，返回登陆状态

backup模块



```
BackUpService
    backUp(String): void
    load(String): void
    getNames(): List<String>
```

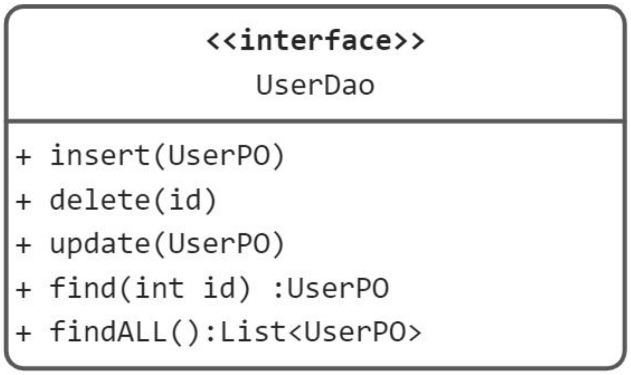


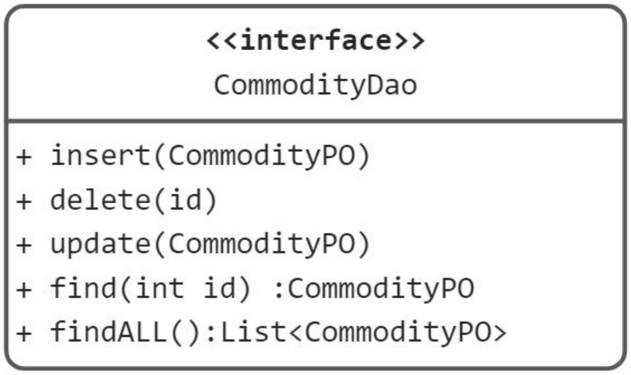
```
BackUpController
    BackUpController(BackUpService)
    backup(String): Response
    load(String): Response
    getName(): Response
    backUpService: BackUpService
```

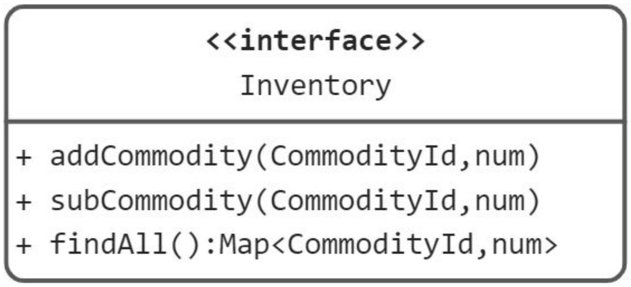
5.10数据层的分解

数据层主要给业务逻辑层提供数据访问服务，包括对于持久化数据的增删改查。由于持久化数据的保存可能存在多种形式：Txt文件、序列化文件、数据库等，所示抽象了数据服务。数据层模块的描述具体如图13所示。

图13 数据层模块的描述






<<interface>>
Inventory
+ addCommodity(CommodityId,num)
+ subCommodity(CommodityId,num)
+ findAll():Map<CommodityId,num>

5.4.1数据层模块的职责

数据层模块的职责如表11所示。

模块

表12 数据层模块

模块	职责
UserDao	用户相关
CommodityDao	商品相关
InventoryDao	库存相关
FinanceDao	金额相关
HRDao	员工管理相关
ReciptDao	单据管理相关

5.4.2数据层模块的接口规范

数据层模块的接口规范如表12所示。

UserDao

接口	语法	前置条件	后置条件
UserDao.createClient	int createUser(UserPO user)	user非空	返回是否创建成功
UserDao.findById	UserPO findById(int Id)	ID非空	返回查找到的UserPO对象，若为null则代表无
UserDao.findAll	List<UserPO> findAll()	无	以列表的形式返回所有user
UserDao.deleteById	int deleteById(int Id)	ID非空	返回是否删除成功

CommodityDao

接口	语法	前置条件	后置条件
CommodityDao.createCommodity	int createCommodity(CommodityPO commodity)	commodity非空	返回是否创建成功
CommodityDao.updateById	int updateByIdCommodityPO commodity)	commodity非空	返回是否修改成功
CommodityDao.findById	CommodityPO findById(int id)	id非空	返回查找到的commodity对象，若为null则代表无
CommodityDao.findAll	List<CommodityPO> findAll()	无	以列表的形式返回所有commodity
CommodityDao.deleteById	int deleteById(int Id)	ID非空	返回是否删除成功

InventoryDao

接口	语法	前置条件	后置条件
InventoryDao.addCommodity	int addCommodity(int id, int num)	id非空，num不为负	返回是否添加成功
InventoryDao.subCommodity	int subCommodity(int id,int num)	id非空，num不为负	返回是否减少成功
InventoryDao.findAll	Map<int id,int num> findAll()	无	返回货物编号和其库存数量

表13 数据层模块的接口规范

提供的服务（供接口）		
userDao.createUser	语法	int createUser(UserDao user);
	前置条件	业务逻辑层调用传入UserDao对象
	后置条件	返回创建用户是否成功

6.信息视角

6.1数据持久化对象

持久化用户对象UserPO的定义如图14所示。

图14 持久化用户对象的定义

持久化用户对象的定义 Java 复制代码

```java
public class UserPO implement Serializable{
    int id;
    String name;
    String password;
    Userrole role;

    public UserPO(int i,String n,String p,UserRole r){
        id = i;
        name = n;
        password = p;
        role = r;
    }

    public String getName(){
        return name;
    }

    public int getID(){
        return id;
    }
    public String getPassword(){
        return password;
    }
    public UserRole getRole(){
        return role;
    }
}
```

6.2Txt持久化格式

Txt数据保持格式以Commodity.txt为例。每行分别对应货号、商品名称、价格、数量。中间用“：”隔开。如下所示：

123：杯子：10：32

456：桌子：20：22

6.3数据库表

数据库中包含 User表、Inventory表、Sales表、Finance表、Employees表、Commodity表、CommodityChange表、Receipt表、SalesRecipt表、Client表、Category表。

其中 Commodity 表的主键 Commodity ID为Inventory表的外键。