

详细设计文档

修改人员	变更时间	变更内容	版本号
马宇森	2022.07.06	完成了 Finance 模块的部分内容以及依赖视角	v0.0.1
苏致成	2022.07.06	完成了 Staff、Salary 模块的内容	v0.0.2
马宇森	2022.07.06	完成 Business 模块的部分内容以及依赖视角	v0.0.3
苏致成	2022.07.07	修改并完善 Salary 模块的接口规范	v0.0.4
郁博文	2022.07.07	完成了 Finance 模块、Business 模块内容	v0.0.5
石睿琪	2022.07.07	完成了 Promotion 模块内容	v0.0.6

1. 引言

1.1 编制目的

本报告详细完成对连锁商店管理系统的详细设计，达到指导后续软件构造的目的，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	进销存管理系统	无

1.3 参考资料

1. IEEE 标准
2. 《软件工程与计算（卷二）软件开发的技术基础》
3. ERP 系统需求规格说明文档
4. ERP 系统软件体系结构文档

2. 产品概述

参考 ERP 系统的用例文档和软件需求规格说明文档中对产品的概括描述。

3. 体系结构设计概述

参考 ERP 系统的概要设计文档中对体系结构设计的概述。

4. 结构视角

4.1 业务逻辑层的分解

业务逻辑层的开发包参见软件体系结构文档。

4.1.1 Financebl 模块

(1) 模块概述

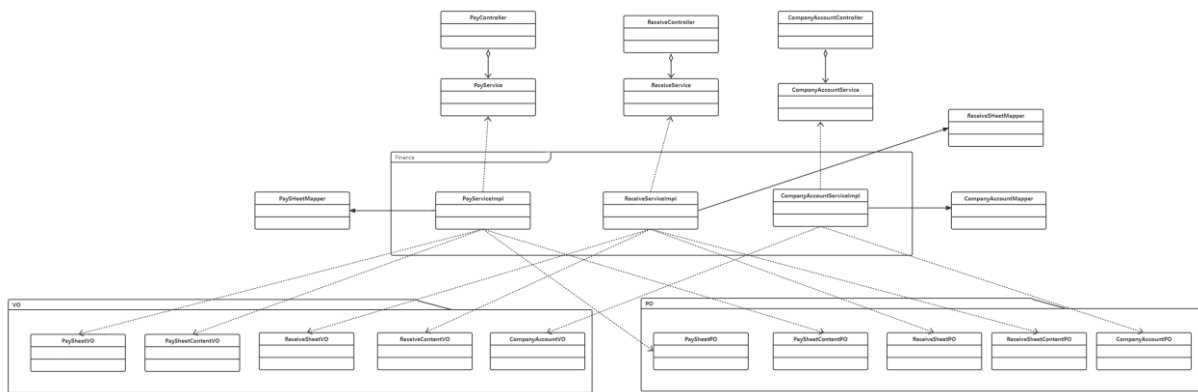
Financebl 模块承担的需求见需求规格说明文档功能需求及相关非功能需求说明文档。

Financebl 模块的职责及接口参见软件系统结构描述文档。

(2) 整体结构

根据体系结构的设计，我们将系统分为了前端展示层、Controller 层、业务逻辑层、数据层。每一层为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们设置了 CompanyAccountService、PayService、ReceiveService 接口，业务逻辑层和数据层之间设置了 Mapper 接口。CompanyAccountPO 是作为公司账户的持久化对象而添加到设计模型中，PaySheetPO 是作为付款单的持久化对象而添加到设计模型中，PayShetContentPO 是作为付款单内容的持久化对象而添加到设计模型中，ReceiveSheetPO 是作为收款单的持久化对象而添加到设计模型中，ReceiveShetContentPO 是作为收款单内容的持久化对象而添加到设计模型中。

Financebl 模块的设计如下图：



各个类的职责：

模块	职责
CompanyAccountController	负责实现公司账户前后端的数据交互
CompanyAccountServiceImpl	负责实现公司账户的增删改查功能
CompanyAccountPO	公司账户增删改查的实现载体，存有 id，名称，余额信息。
CompanyAccountVO	公司账户增删改查的实现载体，存有 id，名称，余额信息。
PayController	负责实现付款单据前后端的数据交互
PayServiceImpl	负责实现付款单据的增删改查、审批功能
PaySheetPO	付款单据增删改查的实现载体，存有单据 id，金额，银行账户等信息。
PaySheetVO	付款单据增删改查的实现载体，存有单据 id，金额，银行账户等信息。
PaySheetContentPO	条目清单增删改查的实现载体，存有条目名称、转账金额、备注等信息。
PaySheetContentVO	条目清单增删改查的实现载体，存有条目名称、转账金额、备注等信息。
ReceiveController	负责实现收款单据前后端的数据交互
ReceiveServiceImpl	负责实现收款单据的增删改查、审批功能
ReceiveSheetPO	收款单据增删改查的实现载体，存有单据 id，客户名称，应收金额等信息。
ReceiveSheetVO	收款单据增删改查的实现载体，存有单据 id，客户名称，应收金额等信息。
ReceiveSheetContentPO	转账列表增删改查的实现载体，存有银行账户、转账金额、备注等信息。
ReceiveSheetContentVO	转账列表增删改查的实现载体，存有银行账户、转账金额、备注等信息。

(3) 模块内部类的接口规范

Finance 模块的各个类的职责和内部类的接口规范已在 ERP 软件体系结构文档中描述。

Finacebl 模块的接口规范

接口名字	语法	前置条件	后置条件
PayController.makePayOrder	makePayOrder(UserVO userVO, PaySheetVO paySheetVO)	无	将付款单信息传递给 PayService
PayController.showSheetByState	showSheetByState(PaySheetState state)	需查找的单据已被建立	从 PayService 获取对应状态的单据
PayController.approval	approval(String paySheetId, PaySheetState state)	需审批的单据已被建立	将审批信息传递给 PayService
PayController.findBySheetId	findBySheetId(String id)	需查找的单据已被建立	从 PayService 获取对应 ID 的单据
ReceiveController.makeReceiveOrder	makeReceiveOrder(UserVO userVO, ReceiveSheetVO receiveSheetVO)	无	将收款单信息传递给 ReceiveService
ReceiveController.showSheetByState	showSheetByState(ReceiveSheetState state)	需查找的单据已被建立	从 ReceiveService 获取对应状态的单据

ReceiveController.ap proval	approval(String receiveSheetId,Receiv eSheetState state)	需审批的单据已被建 立	将审批信息传递给 ReceiveService
ReceiveController.fin dBySheetId	findBySheetId(String id)	需查找的单据已被建 立	从 ReceiveService 获 取对应 ID 的单据
CompanyAccountCo ntroller.addCompany Account	addCompanyAccoun t(CompanyAccountV O companyAccountVO)	无	将账户信息传给 CompanyAccountSer vice
CompanyAccountCo ntroller.updateComp anyAccount	updateCompanyAcc ount(CompanyAccou ntVO companyAccountVO)	更新的账户已经存在	将更新的账户信息传 给 CompanyAccountSer vice
CompanyAccountCo ntroller.deleteCompa nyAccount	deleteCompanyAcco unt(Integer id)	删除的账户已经存在	CompanyAccountSer vice 删除对应账户
CompanyAccountCo ntroller.queryAllCom panyAccounts	queryAllCompanyAcc ounts()	无	从 CompanyAccountSer vice 获取所有账户
CompanyAccountCo ntroller.findCompany AccountByName	findCompanyAccoun tByName(String name)	name 对应的账户存 在	从 CompanyAccountSer vice 中获取对应名称 的账户

Finacebl 模块需要的服务

接口名字

PayService.makePaySheet

PayService.getPaySheetByState

服务

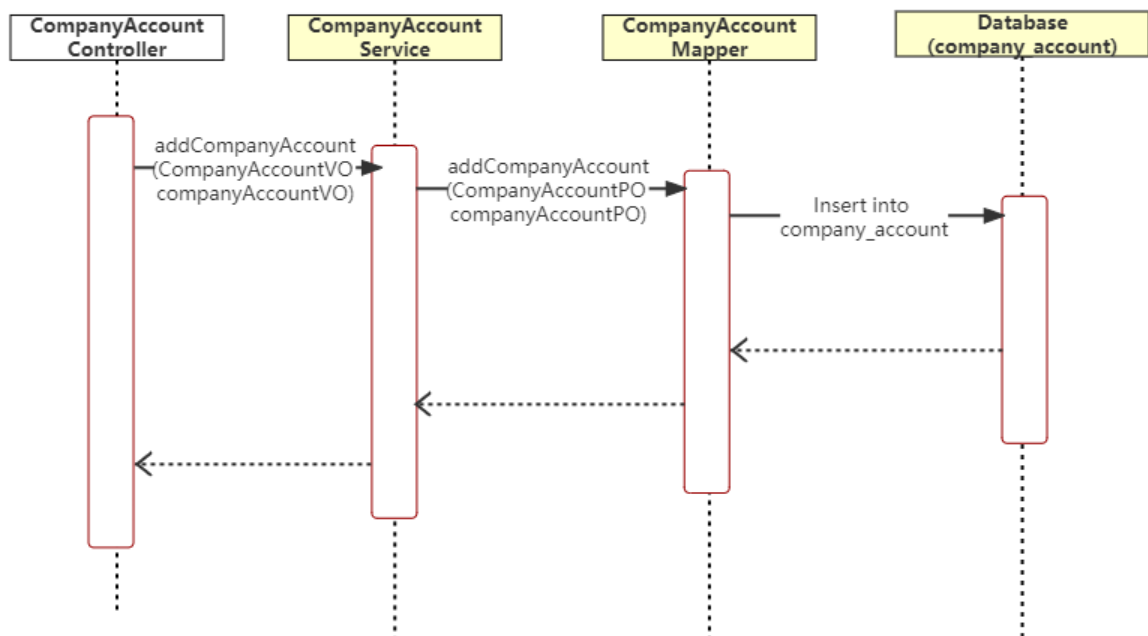
生成一个付款单据

根据单据状态获取对应的付款单据

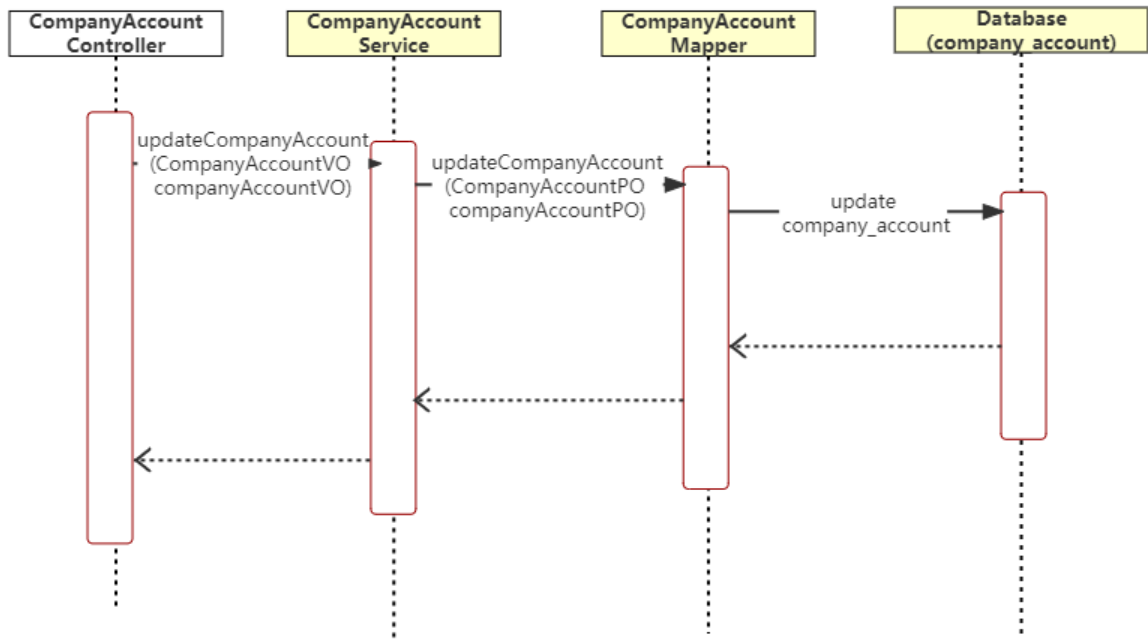
PayService.approval	审批单据
PayService.getPaySheetById	根据单据 ID 获取对应的付款单据
ReceiveService.makeReceiveSheet	生成一个收款单据
getReceiveSheetByState	根据单据状态获取对应的收款单据
ReceiveService.approval	审批单据
ReceiveService.getReceiveSheetById	根据单据 ID 获取对应的收款单据
CompanyAccountService.addCompanyAccount	增加公司账户
CompanyAccountService.updateCompanyAccount	更新公司账户
CompanyAccountService.deleteCompanyAccount	删除公司账户
CompanyAccountService.queryAllCompanyAccounts	查询所有公司账户
CompanyAccountService.findOneByName	根据名字查询公司账户

(4) 业务逻辑层的动态模型

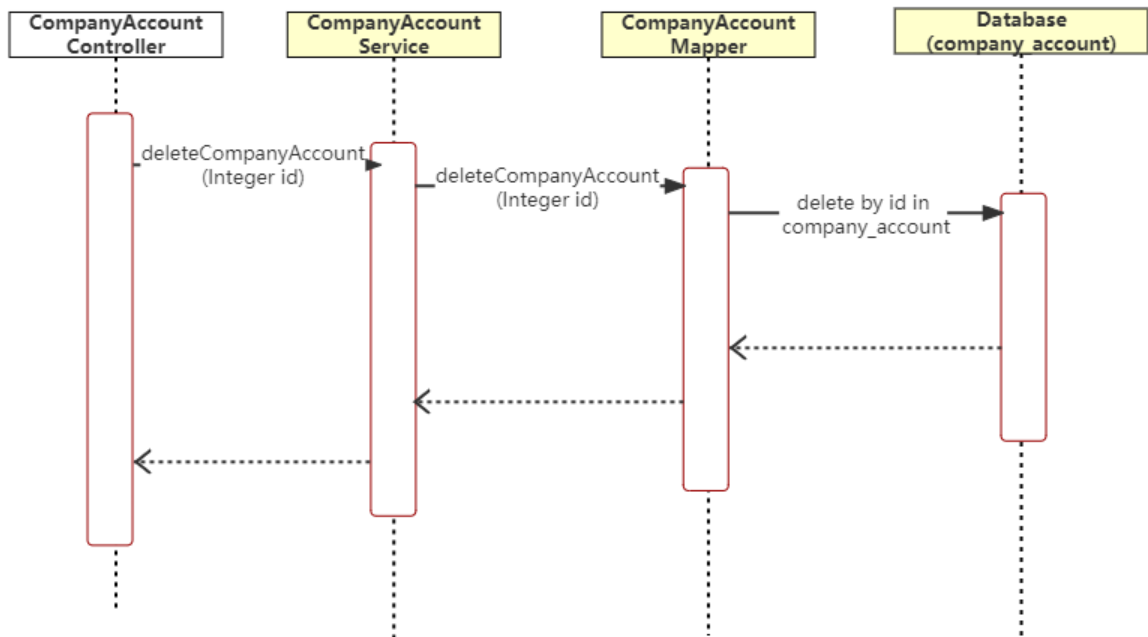
1. 下图表明了 ERP 系统中，Finance.addCompanyAccount 的相关对象之间的协作。



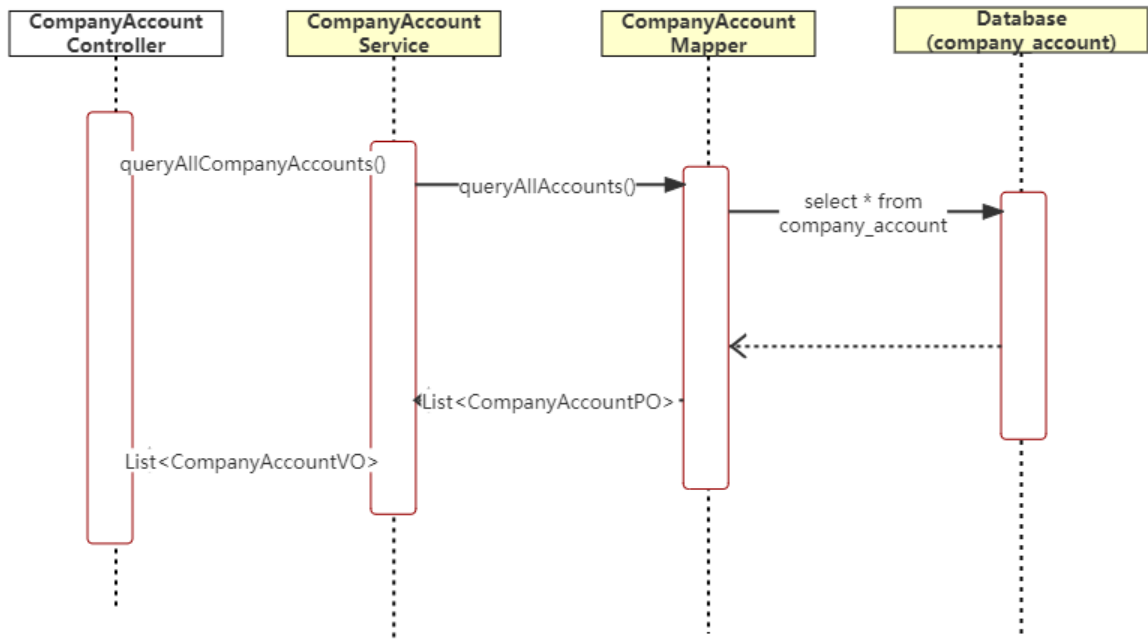
2. 下图表明了 ERP 系统中，Finance.updateCompanyAccount 的相关对象之间的协作。



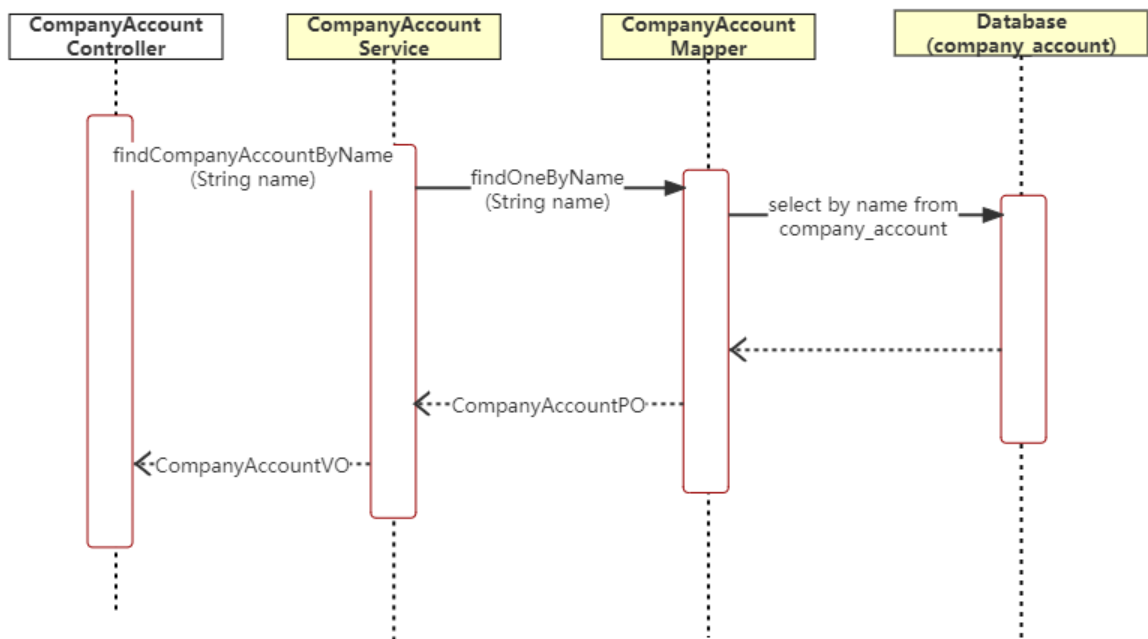
3. 下图表明了 在 ERP 系统中，`Finance.deleteCompanyAccount` 的相关对象之间的协作。



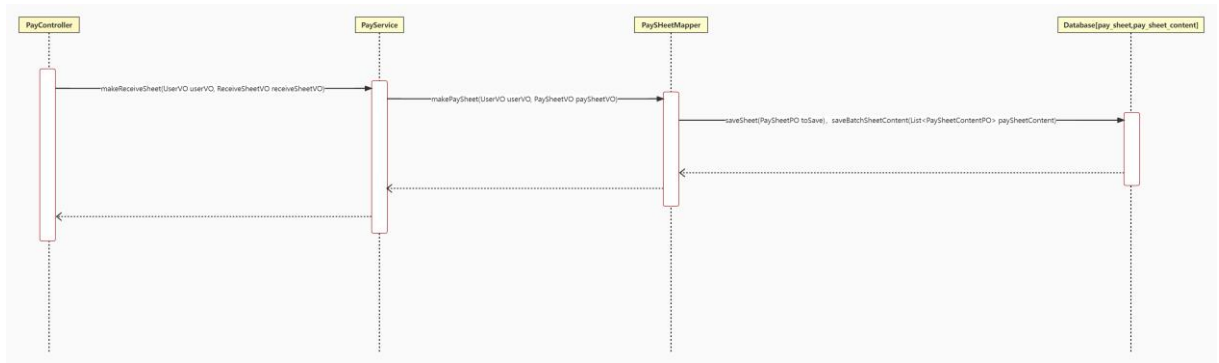
4. 下图表明了 在 ERP 系统中，`Finance.queryAllCompanyAccounts` 的相关对象之间的协作。



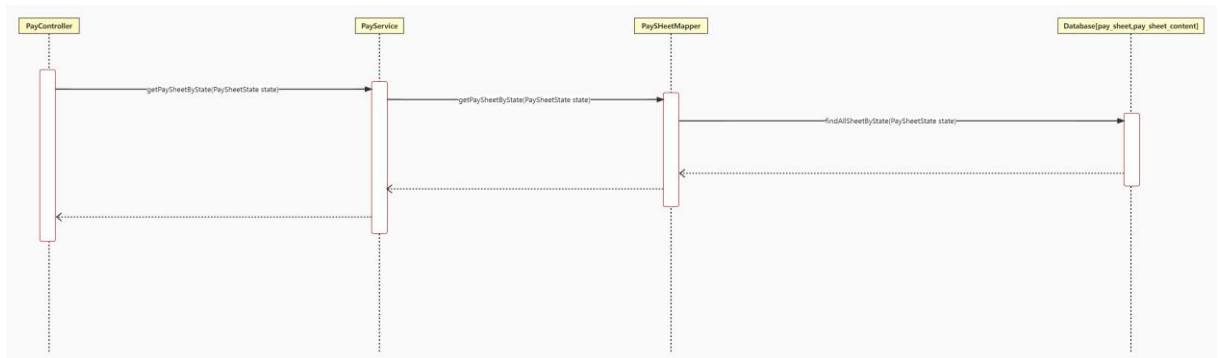
5. 下图表明了 在 ERP 系统中， Finance.findCompanyAccountByName 的相关对象之间的协作。



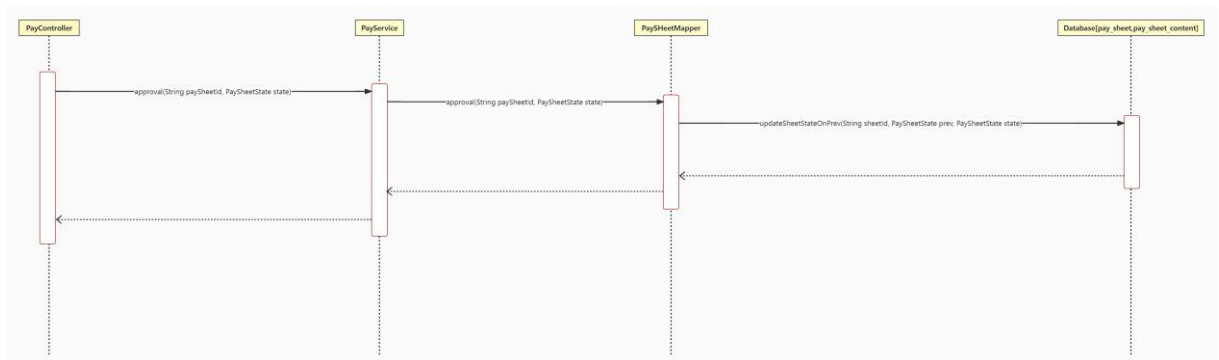
6. 下图表明了 在 ERP 系统中， Finance.makePaySheet 的相关对象之间的协作。



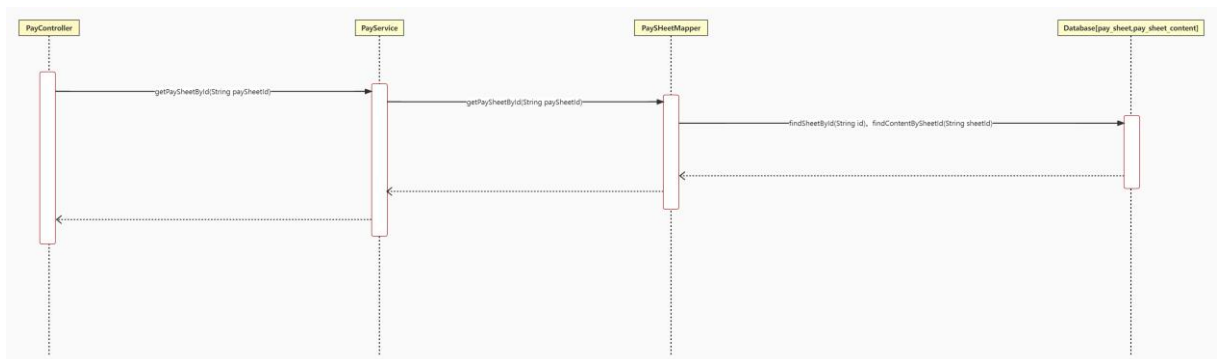
7. 下图表明了 ERP 系统中，Finance.getPaySheetByState 的相关对象之间的协作。



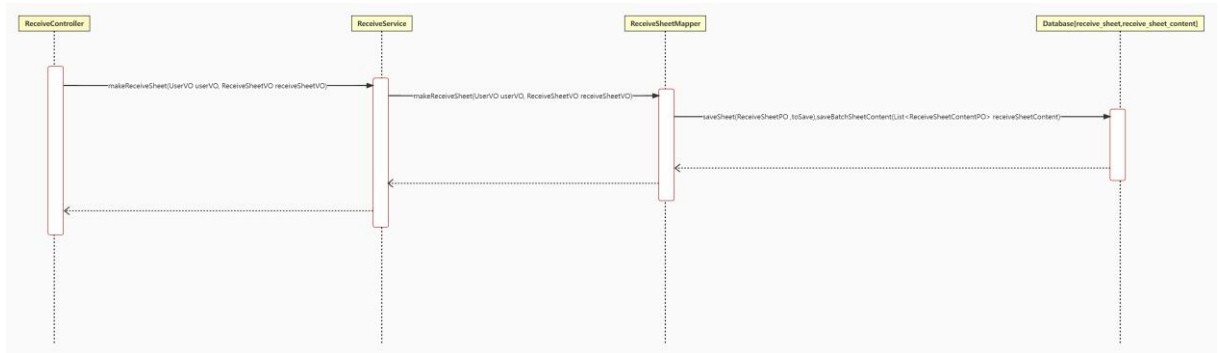
8. 下图表明了 ERP 系统中，Finance.approval 的相关对象之间的协作。



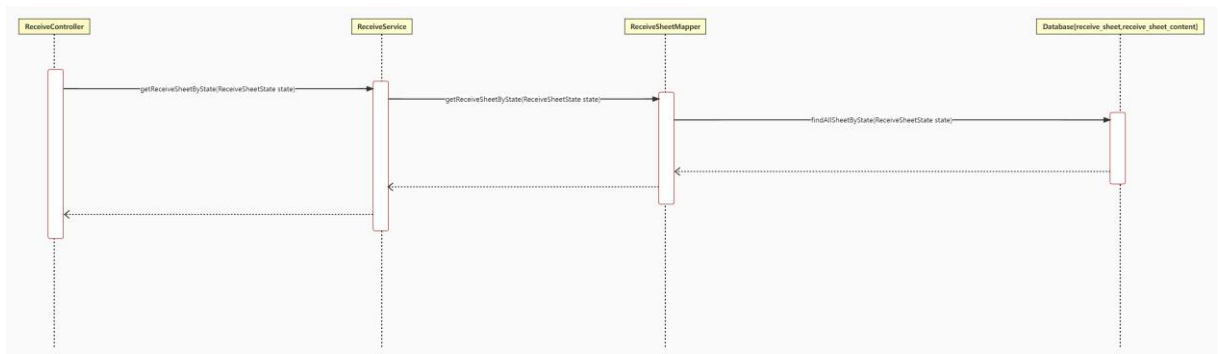
9. 下图表明了 ERP 系统中，Finance.getPaySheetById 的相关对象之间的协作。



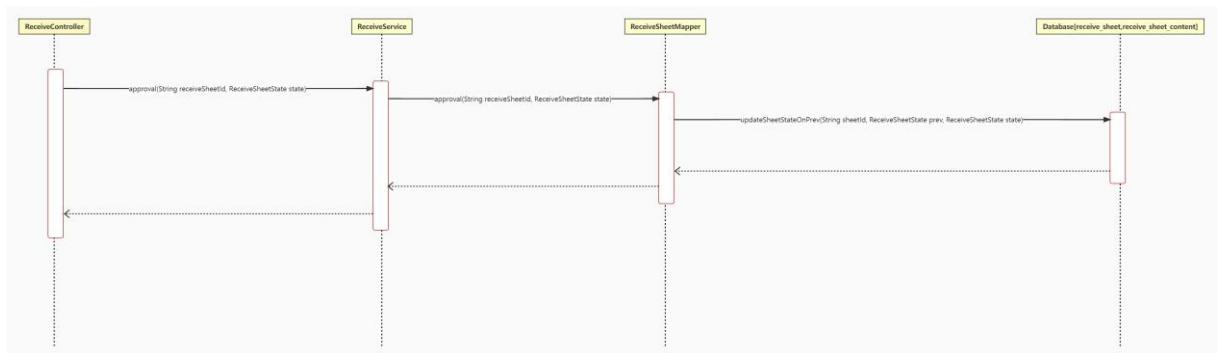
10. 下图表明了 ERP 系统中，Finance.makeReceiveSheet 的相关对象之间的协作。



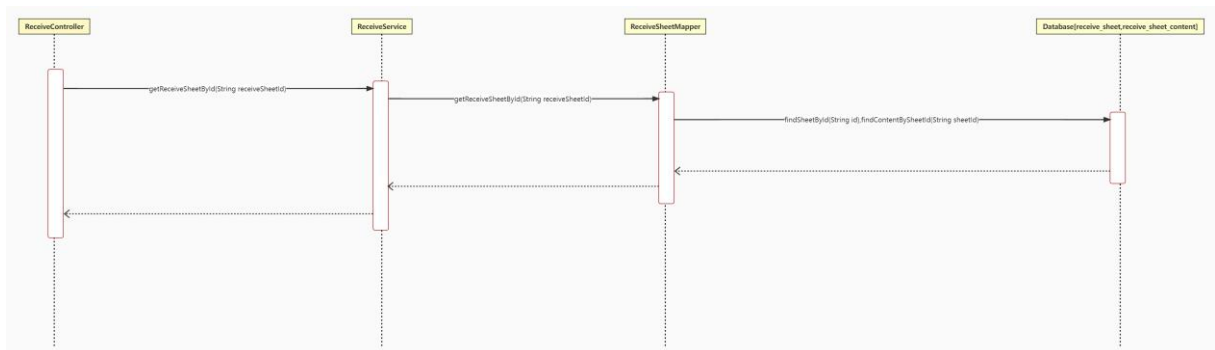
11. 下图表明了 在 ERP 系统中，Finance.getReceiveSheetByState 的相关对象之间的协作。



12. 下图表明了 在 ERP 系统中，Finance.approval 的相关对象之间的协作。



13. 下图表明了 在 ERP 系统中，Finance.getReceiveSheetById 的相关对象之间的协作。



(5) 业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的控制器委托给不同的领域对象。

4.1.2 Businessbl 模块

(1) 模块概述

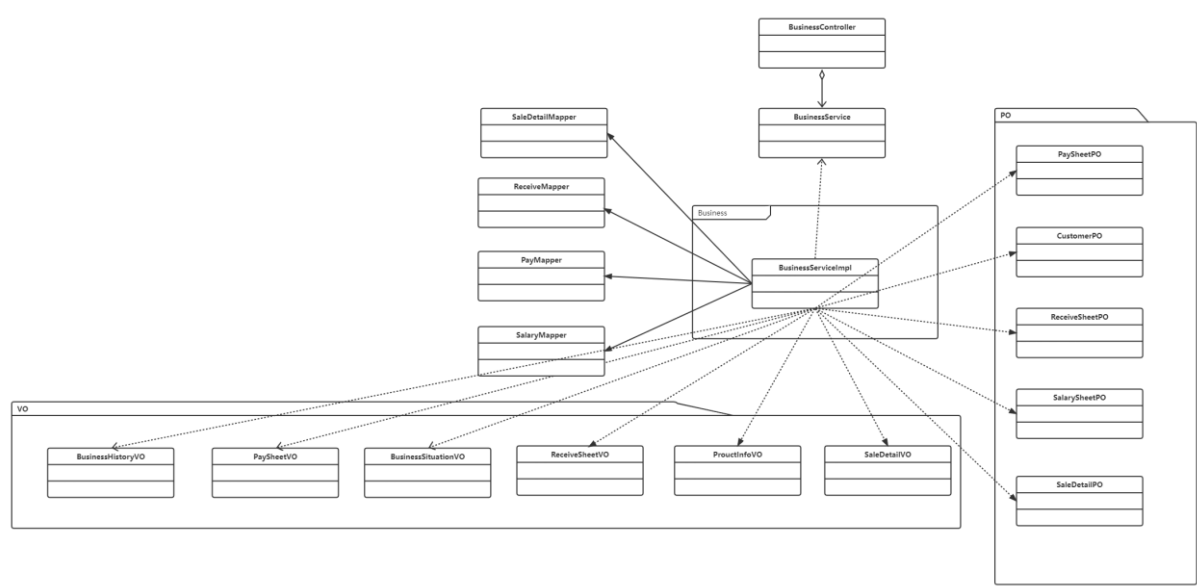
Businssbl 模块承担的需求参见需求规格说明文档功能需求及相关非功能需求说明文档。

Businessbl 模块的职责及接口参见软件系统结构描述文档。

(2) 整体结构

根据体系结构的设计，我们将系统分为了前端展示层、Controller 层、业务逻辑层、数据层。每一层为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们设置了 BusinessService 接口，业务逻辑层和数据层之间设置了 Mapper 接口。SaleDetailPO 是作为销售明细的持久化对象而添加到设计模型中，BusinessSituationPO 是作为经营情况的持久化对象而添加到设计模型中。

Businessbl 模块的设计如下图：



各个类的职责：

模块	职责
BusinessController	负责实现 Business 模块前后端的数据交互
BusinessServiceImpl	负责实现 Business 模块的功能
SaleDetailPO	销售明细的实现载体，存有商品 id，客户 id，单价，数量，总价等信息。
SaleDetailVO	销售明细的实现载体，存有商品名，客户名，单价，数量，总价等信息。
BusinessSituationPO	经营情况的实现载体，存有销售总收入，销

	售折扣，人力成本，商品成本，利润等信息。
BusinessSituationVO	经营情况的实现载体，存有销售总收入，销售折扣，人力成本，商品成本，利润等信息。
BusinessHistoryVO	经营历程的实现载体，保存了各种单据的列表。

(3) 模块内部类的接口规范

Business 模块的各个类的职责和内部类的接口规范已在 ERP 软件体系结构文档中描述。

Businessbl 模块的接口规范

接口名字	语法	前置条件	后置条件
BusinessService.getBusinessSituation	getBusinessSituation(Date fromDate, Date toDate)	无	返回对应的经营情况表
BusinessService.getSaleDetail	List<SaleDetailVO> getSaleDetail()	无	返回所有销售明细
BusinessService.getBusinessHistory	getBusinessHistory	无	返回经营历程

Businessbl 模块需要的服务

服务名	服务
PaySheetMapper.findAllSheetByDate	根据给定日期寻找这段时间内的单据
ReceiveSheetMapper.findAllSheetByDate	根据给定日期寻找这段时间内的单据

SalarySheetMapper.findAllSheetByDate

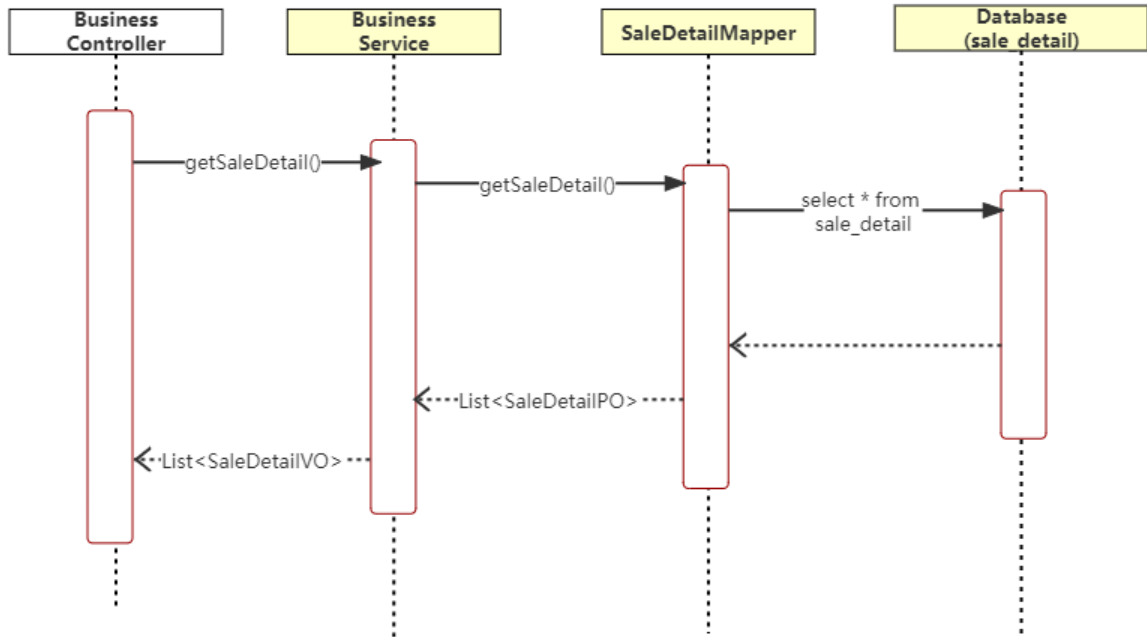
根据给定日期寻找这段时间内的单据

SaleDetailMapper.getSailDetail

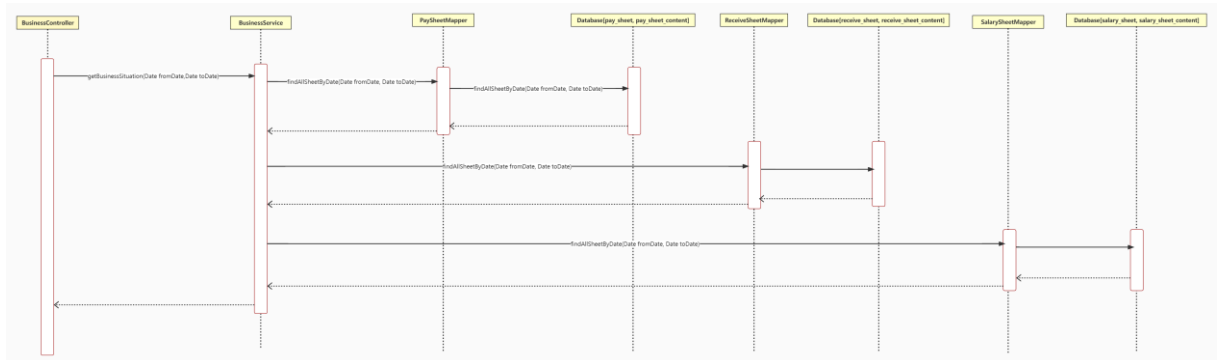
返回所有销售明细

(4) 业务逻辑层的动态模型

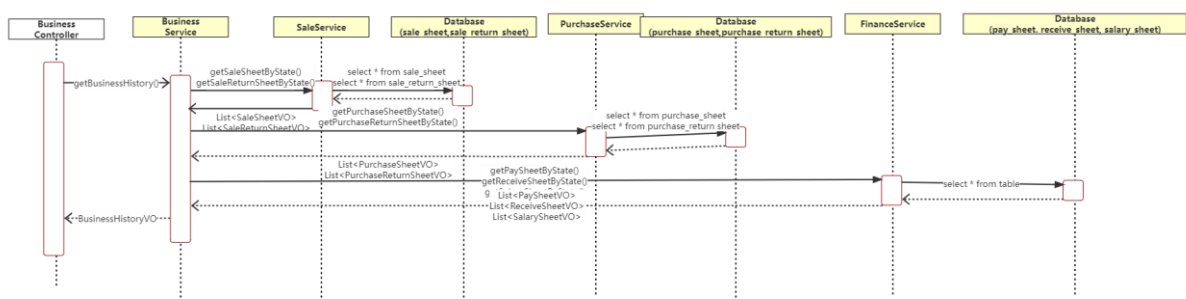
1. 下图表明了 ERP 系统中，Business.getSaleDetail 的相关对象之间的协作。



2. 下图表明了 ERP 系统中，Business.getBusinessSituation 的相关对象之间的协作。



3. 下图表明了 ERP 系统中，Business.getBusinessHistory 的相关对象之间的协作。



(5) 业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的控制器委托给不同的领域对象。

4.1.3 Staffbl 模块

(1) 模块概述

Staffbl 模块承担的需求参见需求规格说明文档功能需求及相关非功能需求说明文档。

Staffbl 模块的职责及接口参见软件系统结构描述文档。

(2) 整体结构

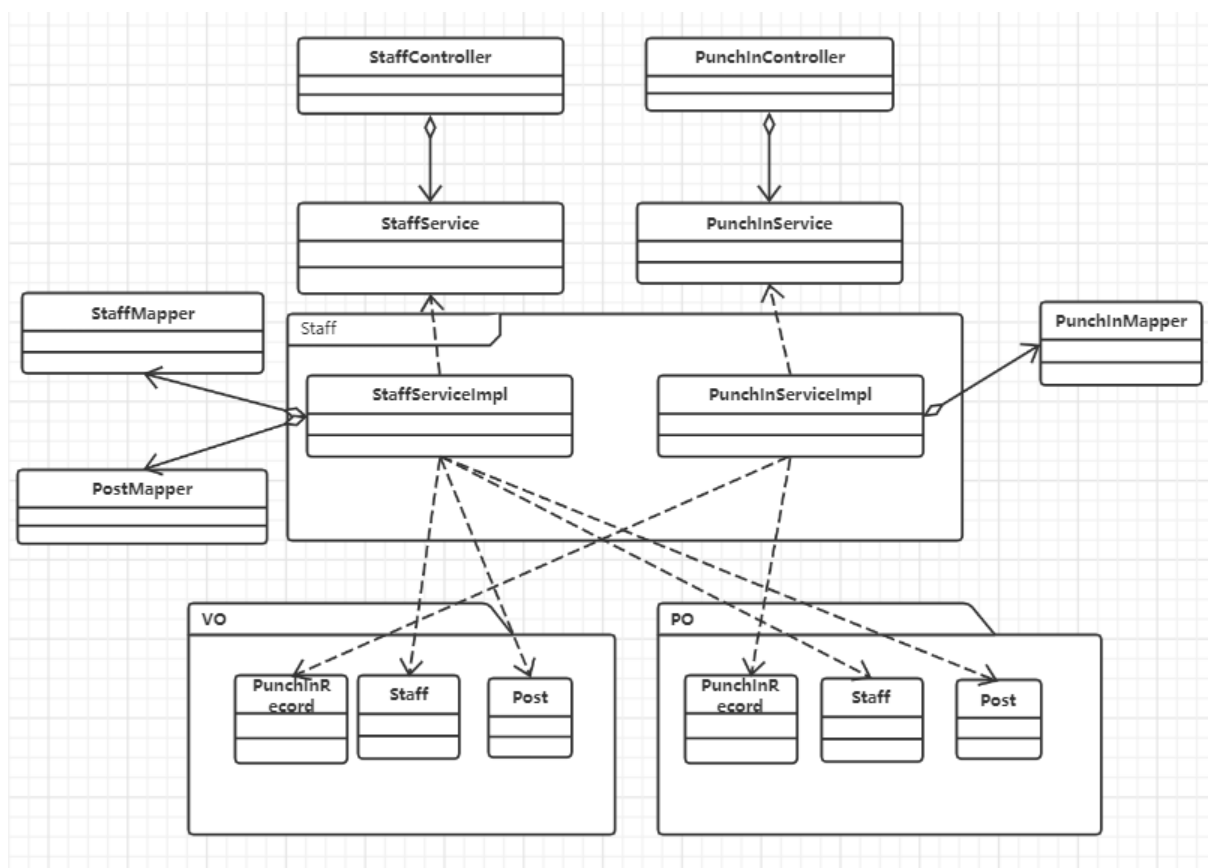
根据体系结构的设计，我们将系统分为了前端展示层、Controller 层、业务逻辑层、数据层。

每一层为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们设置了

StaffService、PunchInService 接口，业务逻辑层和数据层之间设置了 Mapper 接口。

PunchInRecordPO 是作为打卡记录的持久化对象而添加到设计模型中，StaffPO 是作为员工信息持久化对象而添加到设计模型中的，PostPO 是作为岗位信息的持久化对象而添加到设计模型中的。

Staffbl 模块的设计如下图：



各个类的职责：

模块	职责
StaffController	负责实现员工信息前后端的数据交互
StaffServiceImpl	负责实现员工信息的增删改查功能
StaffVO	员工信息增删改查的实现载体，存有员工姓名、生日、电话等信息。
StaffPO	员工信息增删改查的实现载体，存有员工姓名、生日、电话等信息。
PostServiceImpl	负责实现岗位信息的增删改查功能
PostVO	岗位信息增删改查的实现载体，存有岗位名称、基本薪资、岗位薪资等信息。
PostPO	岗位信息增删改查的实现载体，存有岗位名称、基本薪资、岗位薪资等信息。
PunchInController	负责实现打卡信息的前后端的数据交互
PunchInServiceImpl	负责实现打卡信息的增查功能
PunchInVO	打卡信息增查的实现载体，存有员工 ID、打卡时间。
PunchInPO	打卡信息增查的实现载体，存有员工 ID、打

	卡时间。
--	------

(3) 模块内部类的接口规范

Staff 模块的各个类的职责和内部类的接口规范已在 ERP 软件体系结构文档中描述。

Staffbl 模块的接口规范

接口名字	语法	前置条件	后置条件
PunchInController.punchIn	PunchIn(UserVO user)	需员工信息已录入	将打卡信息传递给 PunchInService
PunchInController.findAllPunchInRecordByPeriod	findAllPunchInRecordByPeriod(String beginDateStr,String endDateStr)	需时间格式正确（初始时间小于结束时间）	从 PunchInService 获取所有的打卡信息
PunchInController.findPunchInRecordContentByStaffAndPeriod	findPunchInRecordContentByStaffAndPeriod(int staffId, String beginDateStr, String endDateStr)	需时间格式正确（初始时间小于结束时间）	从 PunchInService 获取某一员工的打卡信息
PunchInController.countByIdAndPeriod	getPaySheetById(int id, Date t1,Date t2)	无	从 PunchInService 获取打卡次数
StaffController.addStaff	addStaff(StaffVO staffVO)	需填写的岗位 ID 已被创建	将员工信息传递给 StaffService
StaffController.queryAll	queryAll()	无	从 StaffService 获取对应状态的员工信息

StaffController.queryAllPost	queryAllPost()	无	从 PostService 获取对应状态的岗位信息
StaffController.deleteStaff	deleteStaff(int staffId)	删除的员工已经存在	StaffService 删除对应员工信息
StaffController.deletePost	deletePost(int postId)	删除的岗位是否有员工与之相关联	StaffService 删除对应岗位信息
StaffController.updatePost	updatePost(PostVO postVO)	更新的岗位已经存在	PostService 更新对应岗位
StaffController.updateStaff	updateStaff(StaffVO staffVO)	更新的员工已经存在	StaffService 更新对应员工信息
StaffController.addPost	addPost(PostVO postVO)	无	将岗位信息传递给 PostService

Staffbl 模块的服务接口

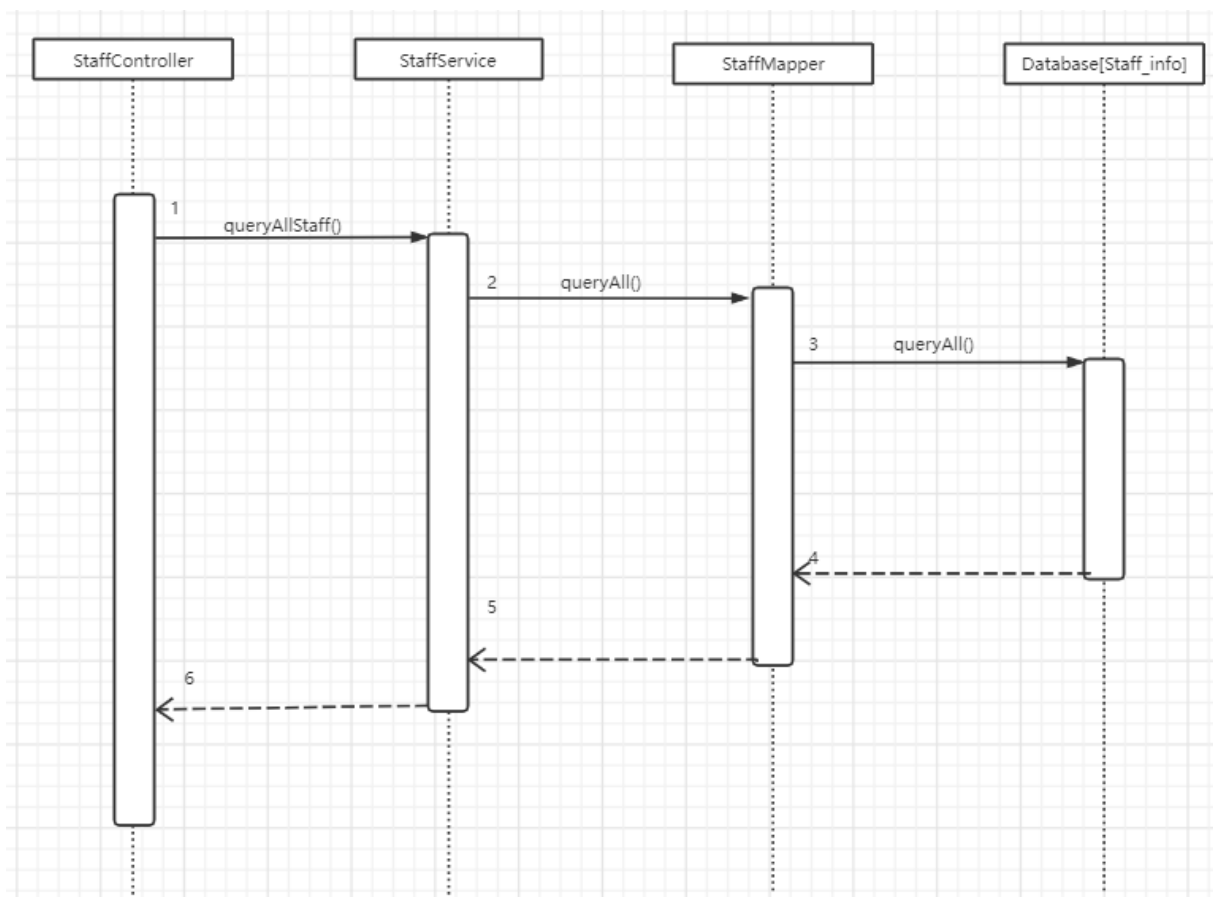
服务名	服务
PunchInService.punchIn	员工打卡
PunchInService.findAllPunchInRecordByPeriod	根据时间段查找打卡记录
PunchInService.findPunchInRecordContentByStaffAndPeriod	根据时间段和员工 ID 查找打卡记录
PunchInService.countByIdAndPeriod	根据时间段和员工 ID 查找打卡次数

StaffService.addStaff
 StaffService.queryAll
 StaffService.queryAllPost
 StaffService.deleteStaff
 StaffService.deletePost
 StaffService.updatePost
 StaffService.updateStaff
 StaffService.addPost
 StaffService.findByStaffId
 StaffService.findCalSalaryMethod
 StaffService.findBasicSalary

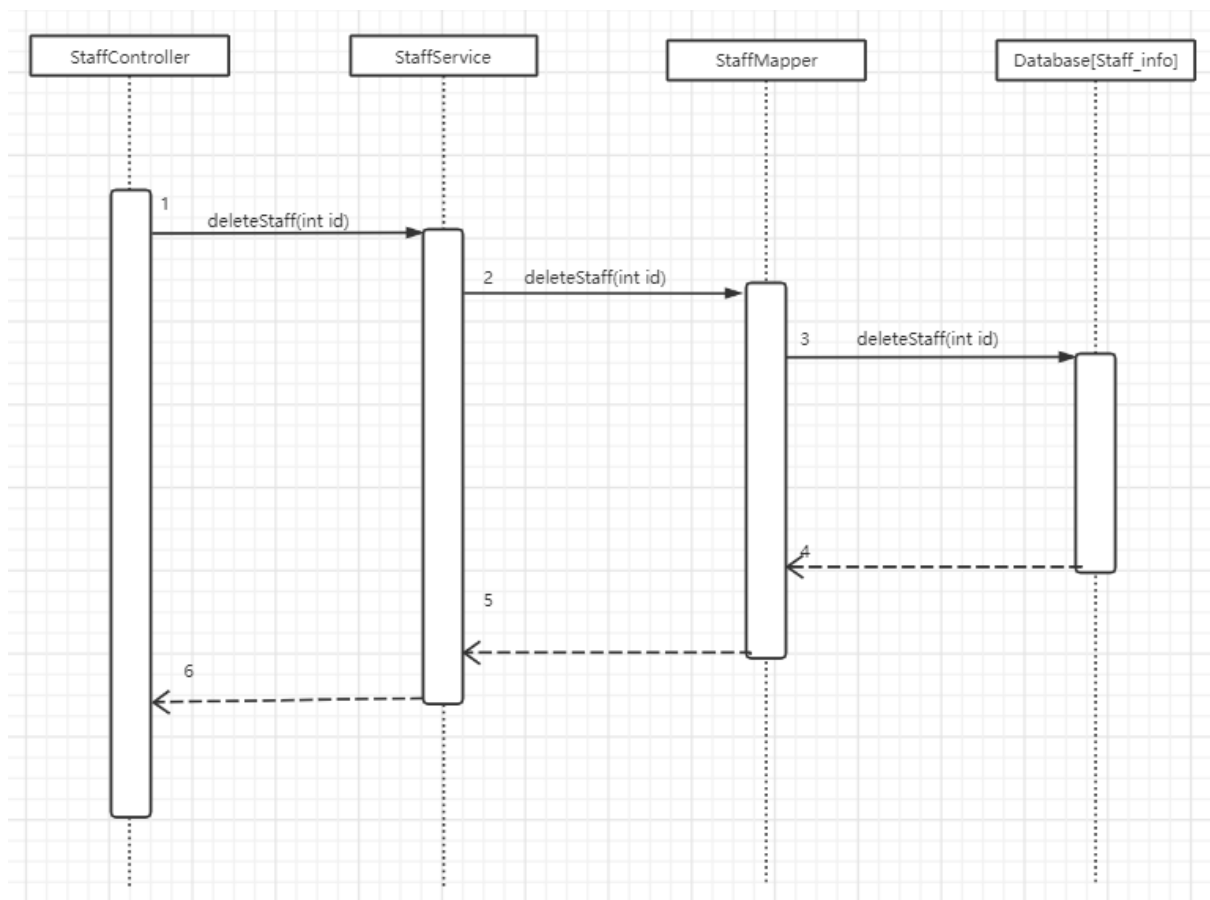
数
 新增员工
 查询所有员工
 查询所有岗位
 删除员工
 删除岗位
 更新岗位
 更新员工
 新增岗位
 根据员工 ID 查找员工
 根据员工 ID 查找薪资计算方法
 根据员工 ID 查找基本薪资

(4) 业务逻辑层的动态模型

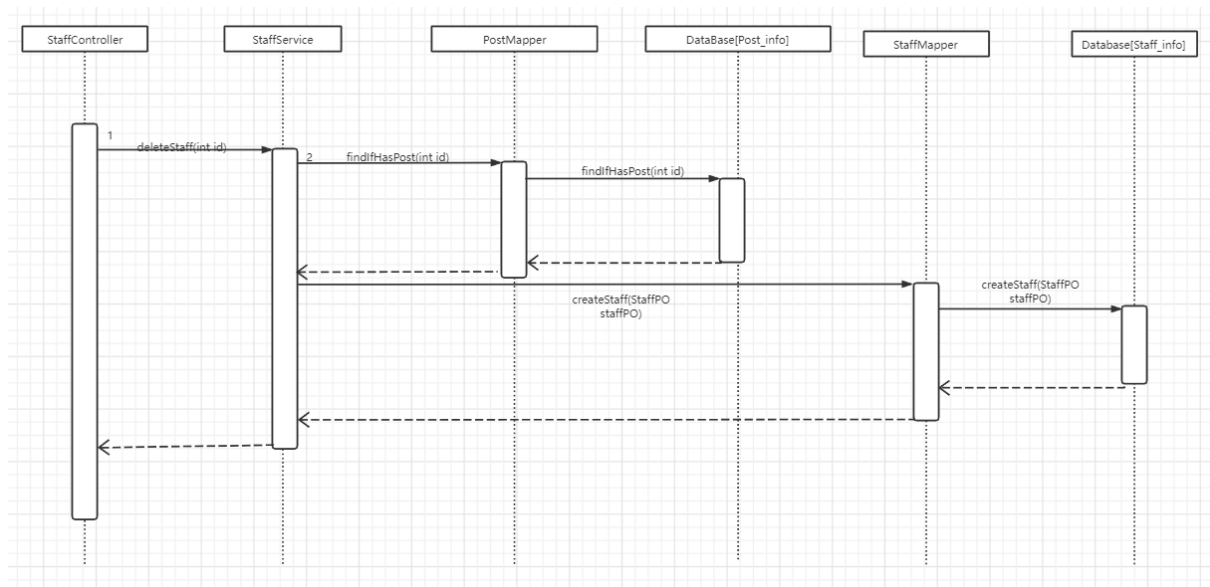
1. 下图表明了 ERP 系统中，Staff.queryAllStaff 的相关对象之间的协作。



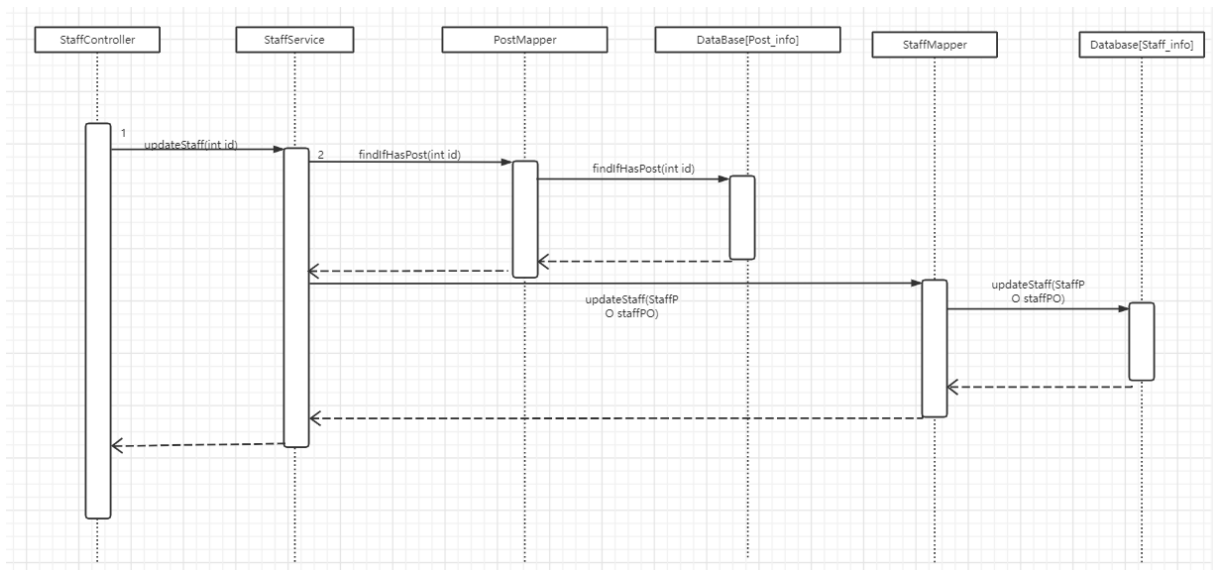
2. 下图表明了 在 ERP 系统中, Staff.deleteStaff 的相关对象之间的协作。



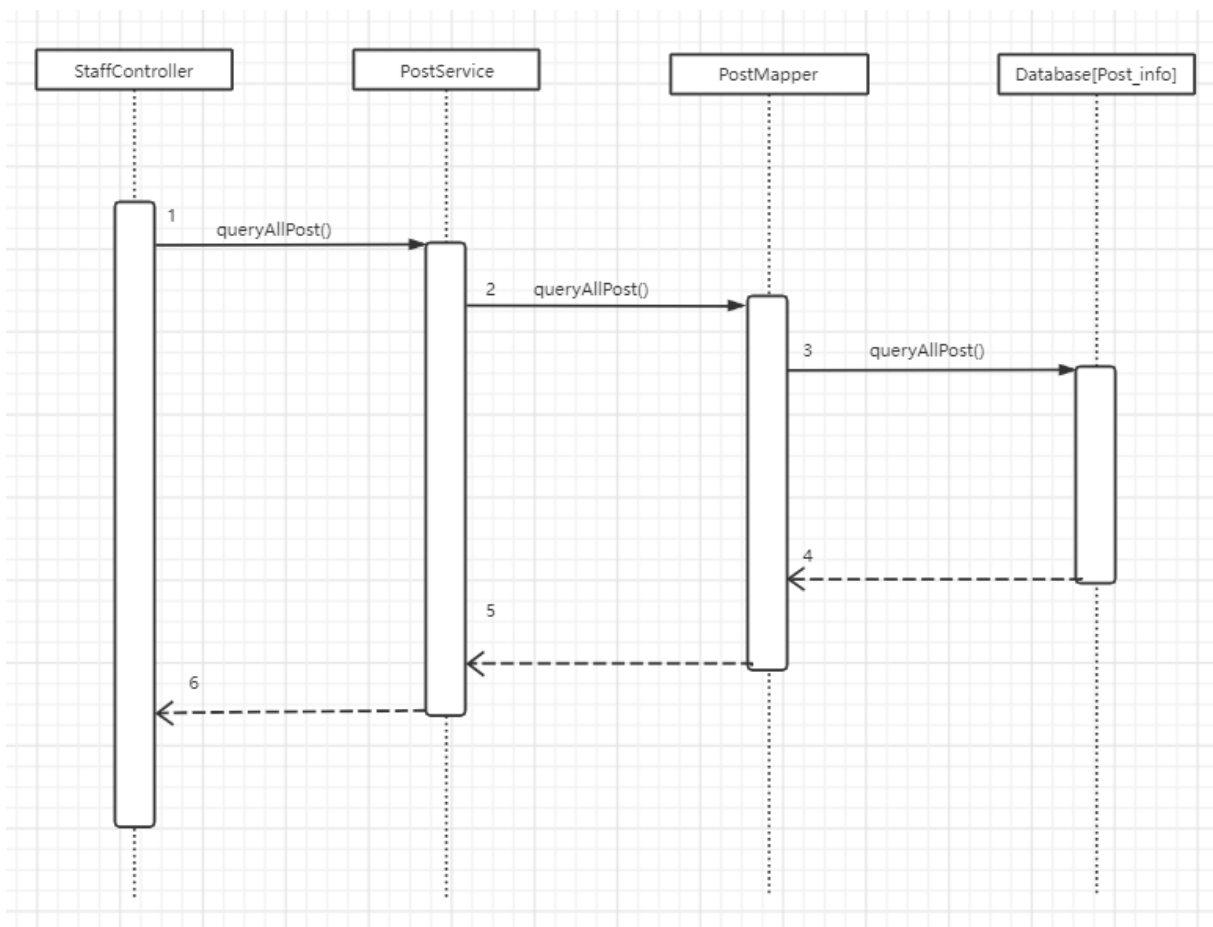
3. 下图表明了 在 ERP 系统中, Staff.addStaff 的相关对象之间的协作。



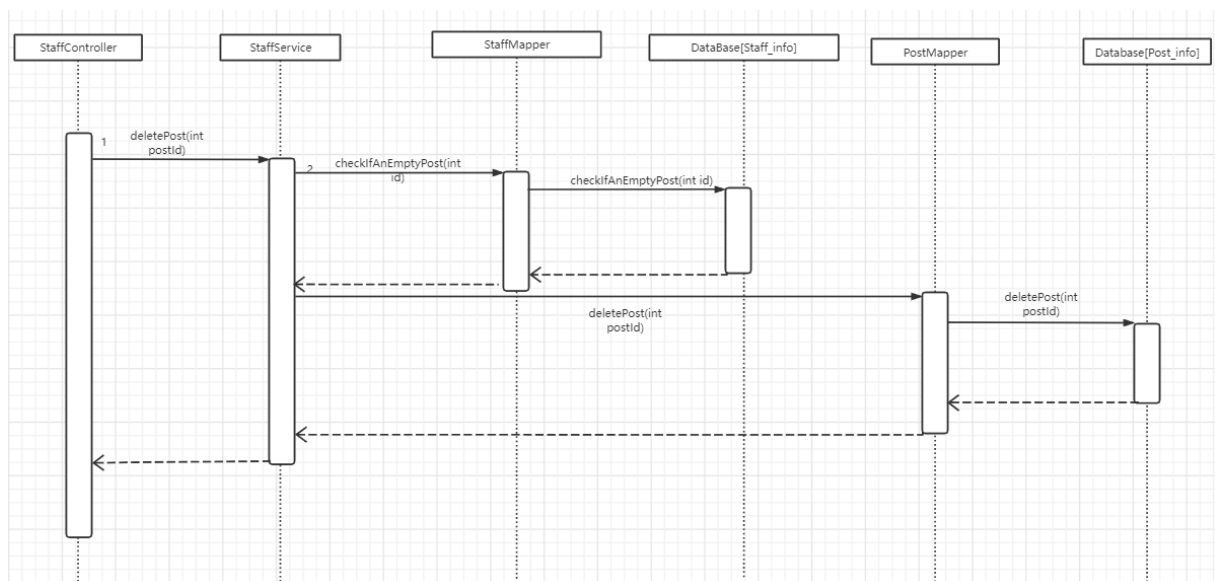
4. 下图表明了 在 ERP 系统中, Staff.updateStaff 的相关对象之间的协作。



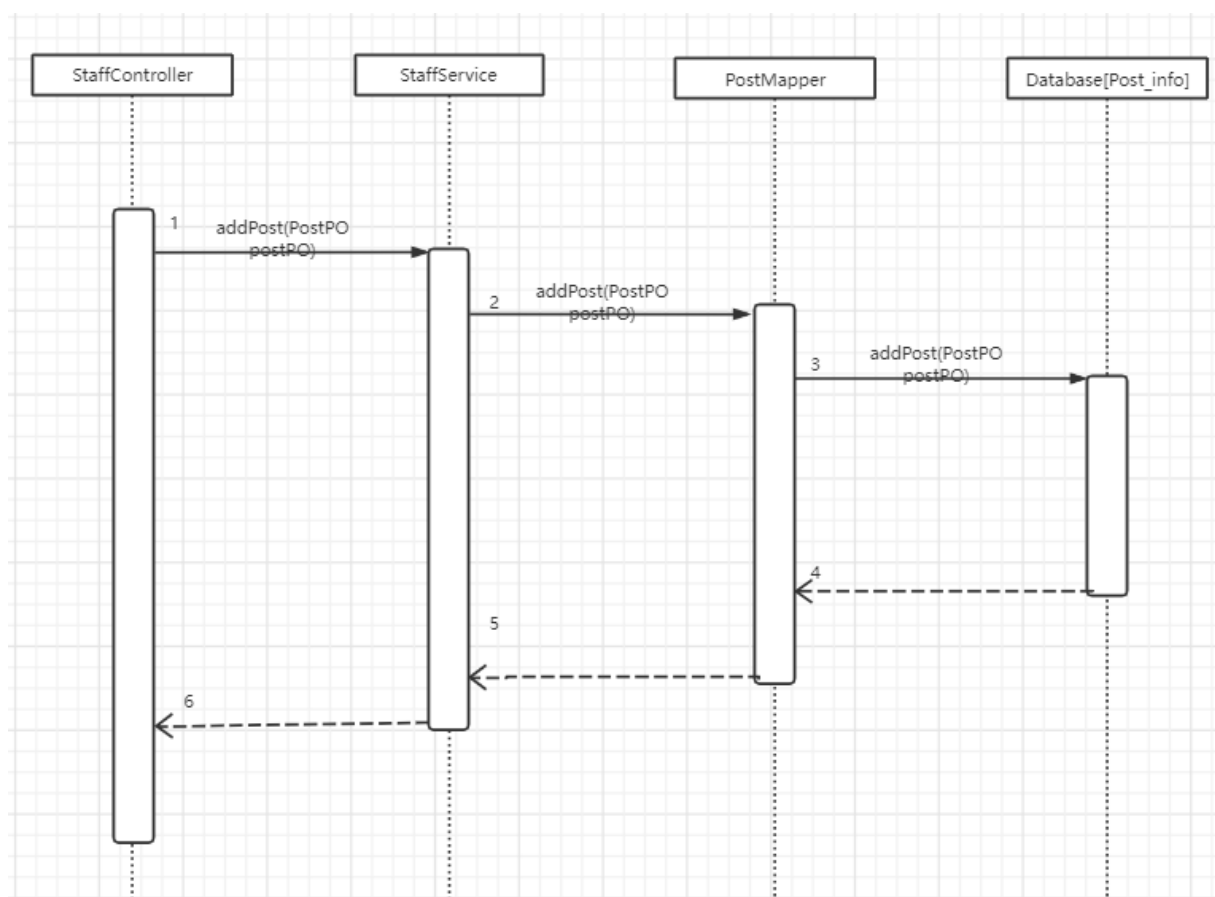
5. 下图表明了 在 ERP 系统中，Staff.queryAllPost 的相关对象之间的协作。



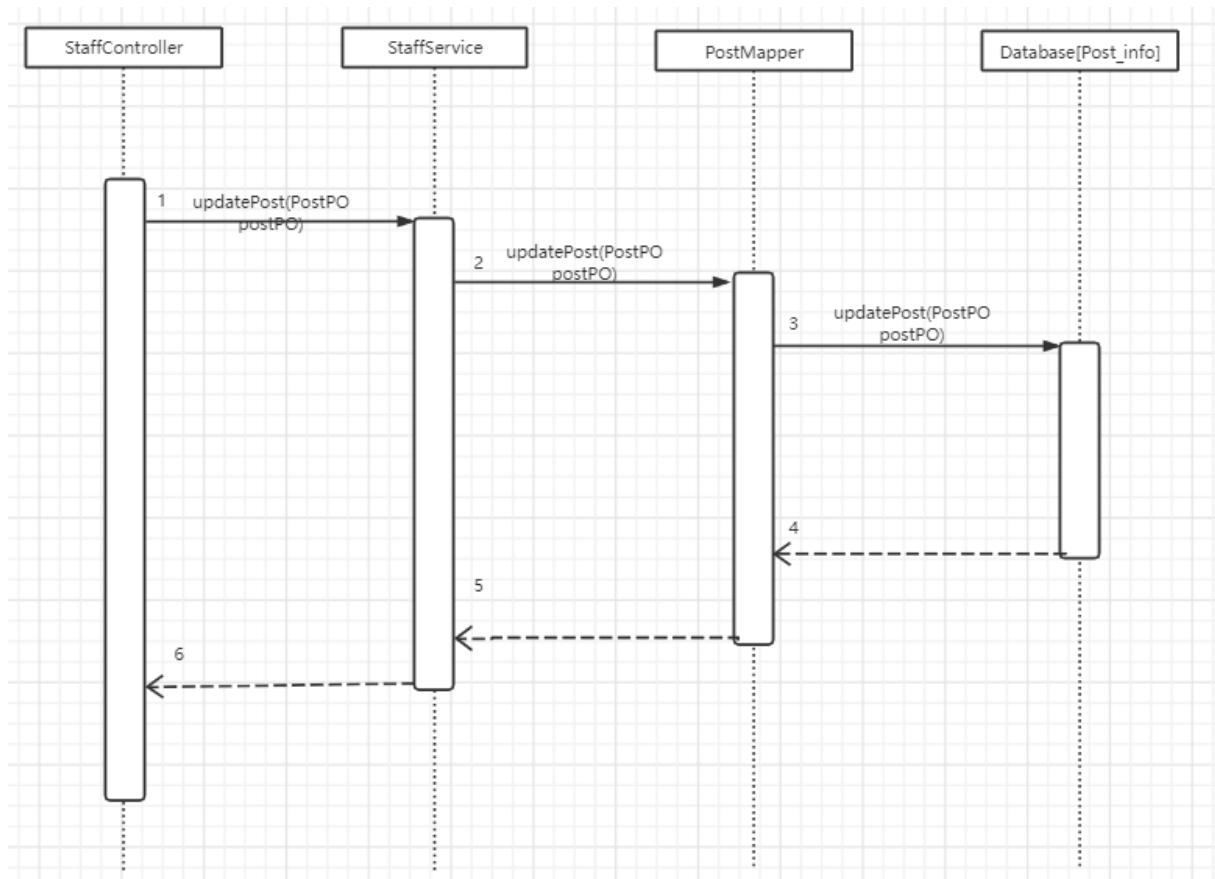
14. 下图表明了 在 ERP 系统中，Staff.deletePost 的相关对象之间的协作。



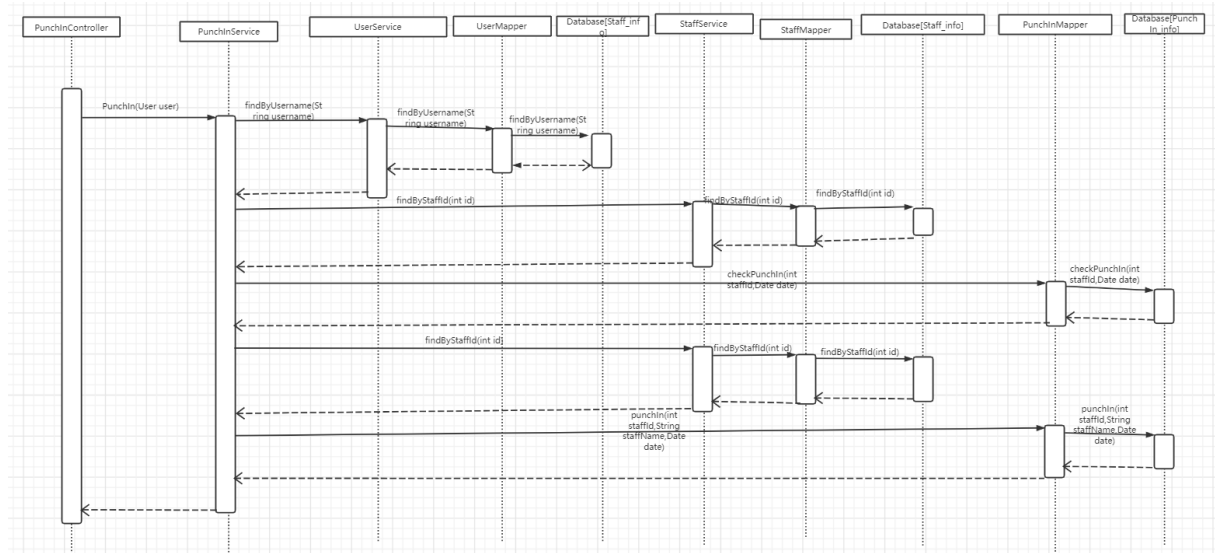
15. 下图表明了 在 ERP 系统中，Staff.addPost 的相关对象之间的协作。



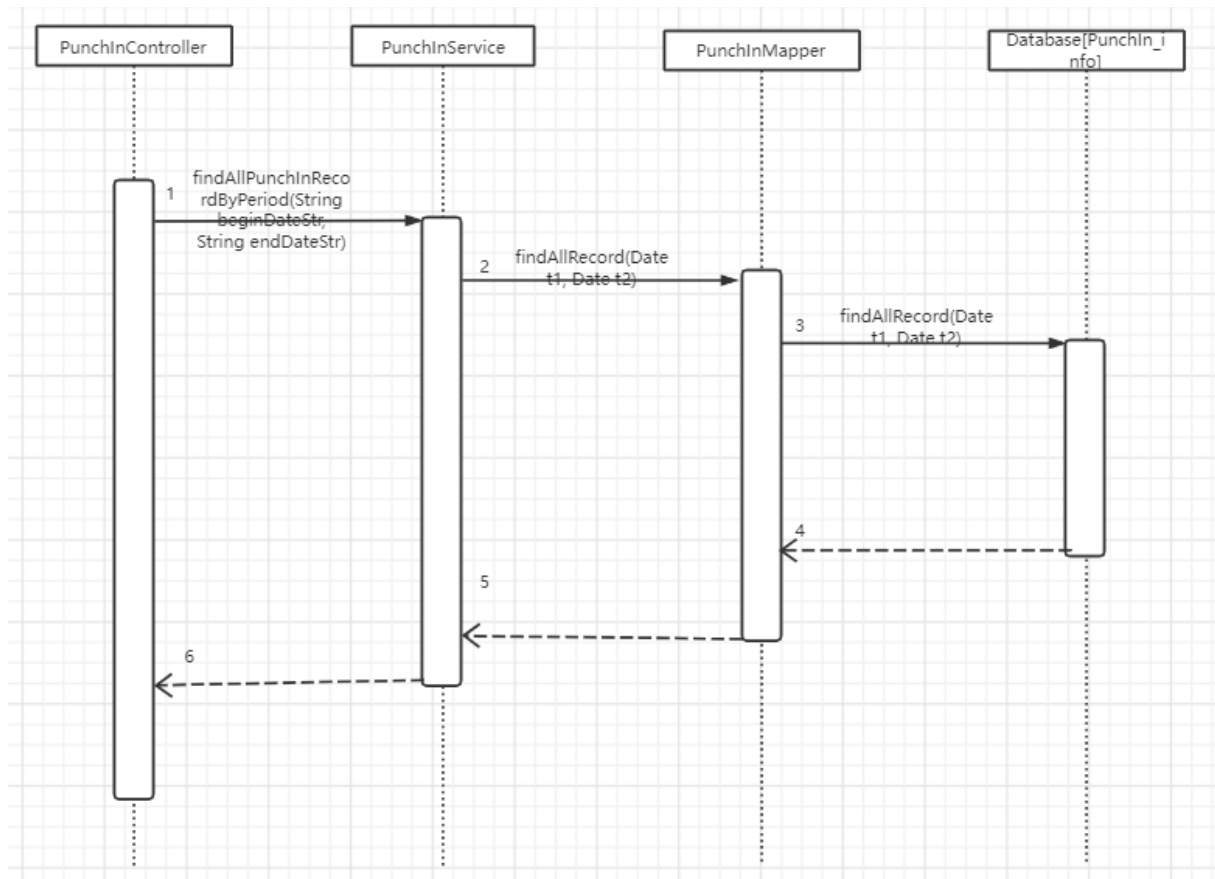
16. 下图表明了 在 ERP 系统中，Staff.updatePost 的相关对象之间的协作。



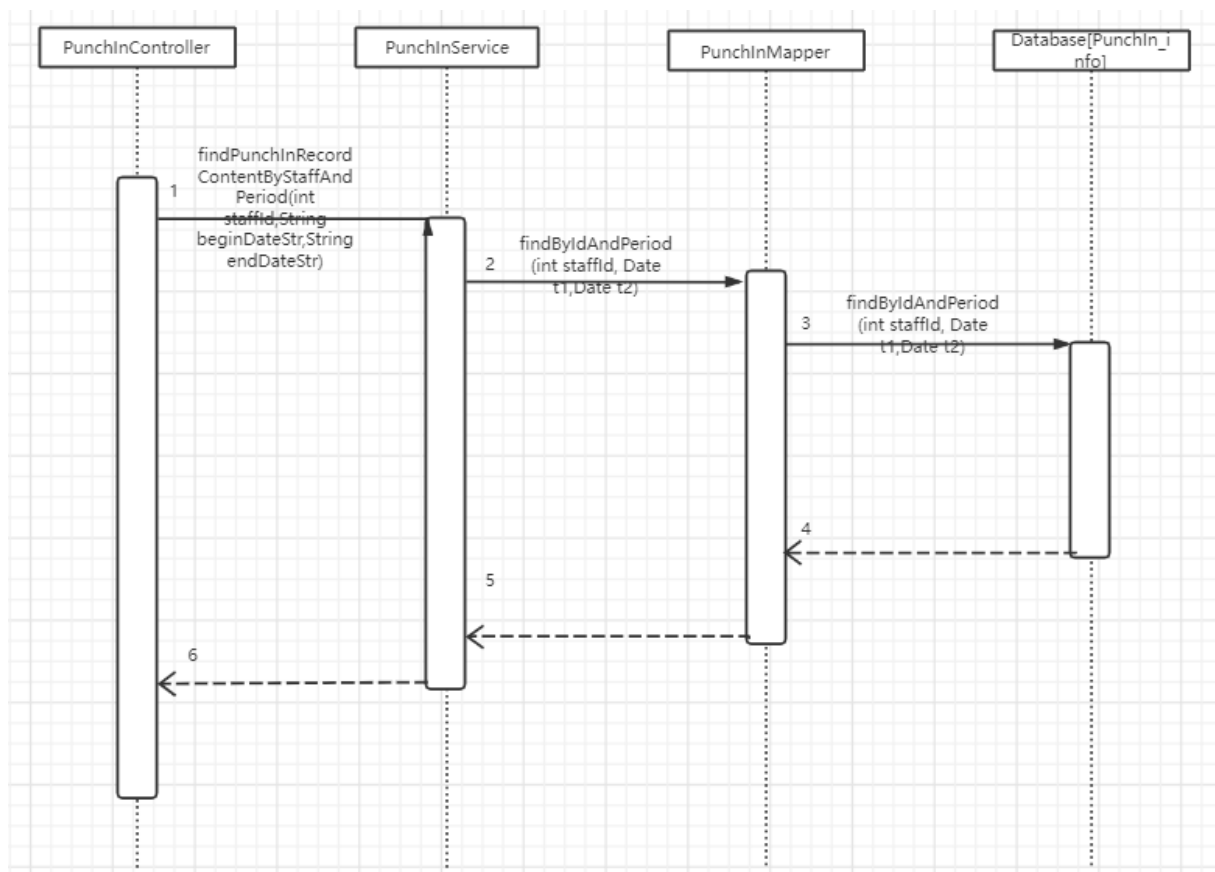
17. 下图表明了 在 ERP 系统中，Staff.PunchIn 的相关对象之间的协作。



18. 下图表明了 在 ERP 系统中，Staff.queryAllRecord 的相关对象之间的协作。



19. 下图表明了 在 ERP 系统中，`Staff.queryAllRecordByStaff` 的相关对象之间的协作。



(5) 业务逻辑层的设计原理

利用委托式风格，每个界面需要访问的业务逻辑由各自的控制器委托给不同的领域对象

4.1.4 Salarybl 模块

(1) 模块概述

Salarybl 模块承担的需求参见需求规格说明文档功能需求及相关非功能需求说明文档。

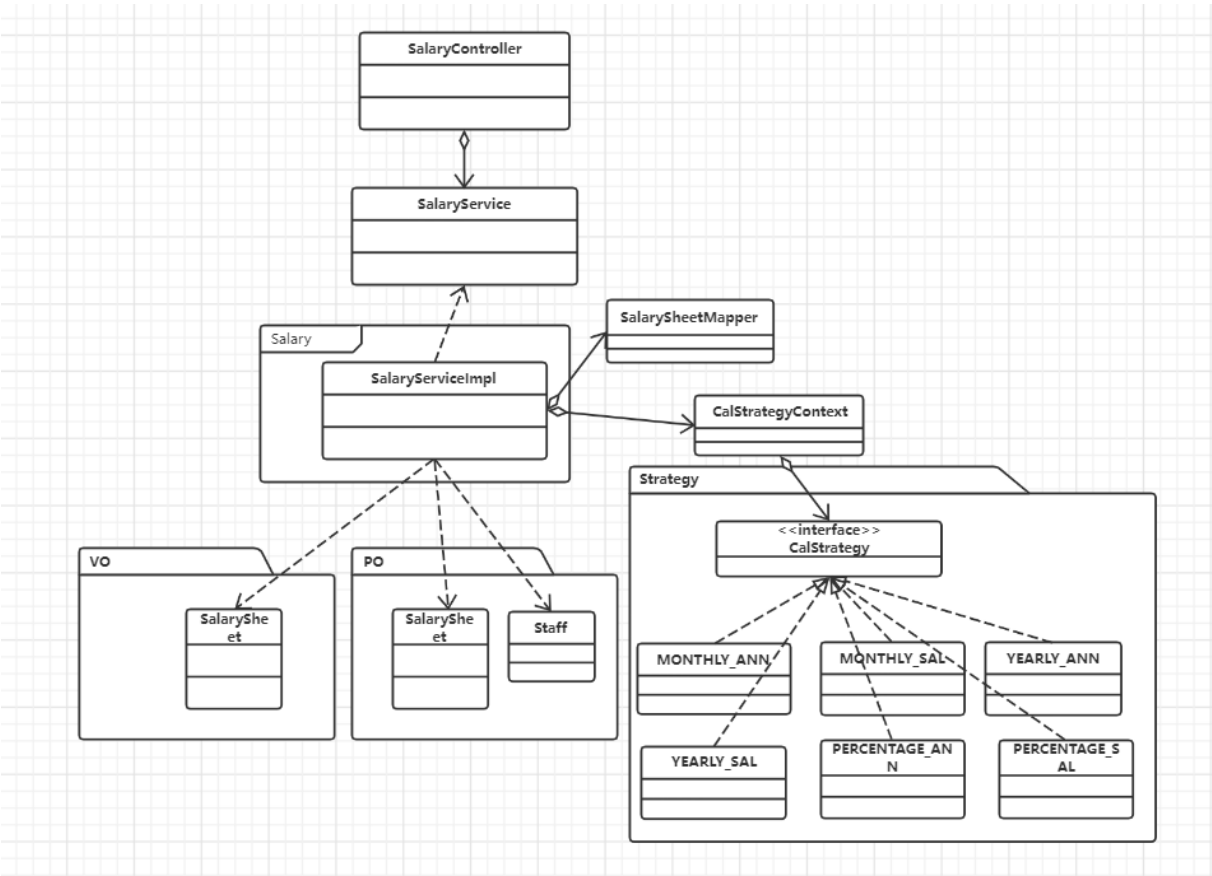
Salarybl 模块的职责及接口参见软件系统结构描述文档。

(2) 整体结构

根据体系结构的设计，我们将系统分为了前端展示层、Controller 层、业务逻辑层、数据层。

每一层为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们设置了 SalaryService 接口，业务逻辑层和数据层之间设置了 Mapper 接口。SalarySheetPO 是作为打卡记录的持久化对象而添加到设计模型中。在该类中我们使用了策略模式，用来实现对月薪策略的灵活变更。

Salarybl 模块的设计如下图：



各个类的职责：

模块	职责
SalaryController	负责实现工资单前后端的数据交互
SalaryServiceImpl	负责实现工资单的增查功能
SalarySheetVO	工资单增查的实现载体，存有 id、员工 id、员工姓名、税前工资、税款、税后工资等。
SalarySheetPO	工资单增查的实现载体，存有 id、员工 id、员工姓名、税前工资、税款、税后工资等。
CalStrategyContext	负责生成相应的策略
CalStrategy	制定工资策略的接口
MONTHLY_ANN	负责实现月薪人员年终奖的增加功能
MONTHLY_SAL	负责实现月薪人员月薪的增加功能
YEARLY_ANN	负责实现年薪人员年终奖的增加功能

MONTHLY_SAL	负责实现年薪人员年薪的增加功能
PERCENTAGE_ANN	负责实现提成制人员年终奖的增加功能
PERCENTAGE_SAL	负责实现提成制人员月薪的增加功能

(3) 模块内部类的接口规范

Salary 模块的各个类的职责和内部类的接口规范已在 ERP 软件体系结构文档中描述。

Salary 模块的接口规范

接口名字	语法	前置条件	后置条件
SalaryController.makeSalary	makeSalary(UserVO userVO)	无	将打卡信息传递给 SalarySheetService
SalaryController.makeAnnual	makeAnnual(UserVO userVO)	年终奖未被制定	将薪资信息传递给 SalarySheetService
SalaryController.showSheetByState	getSalarySheetByState(SalarySheetState state)	无	从 SalarySheetService 获取工资记录
SalaryController.firstApproval	approval(String salarySheetId, SalarySheetState state)	需审批的单据已被建立	将审批信息传递给 SalarySheetService
SalaryController.secondApproval	approval(String salarySheetId, SalarySheetState state)	需审批的单据已被建立	将审批信息传递给 SalarySheetService

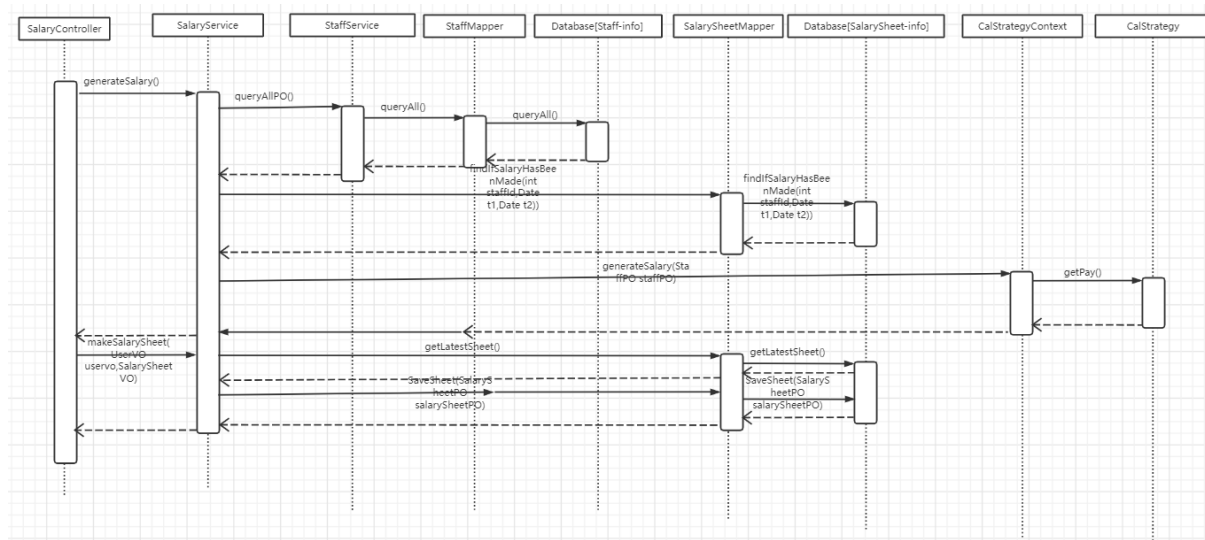
SalaryController.findBySheetId	findBySheetId(String id)	需查找的单据已被建立	从 SalarySheetService 获取对应 ID 的单据
--------------------------------	--------------------------	------------	----------------------------------

Salary 模块的服务接口

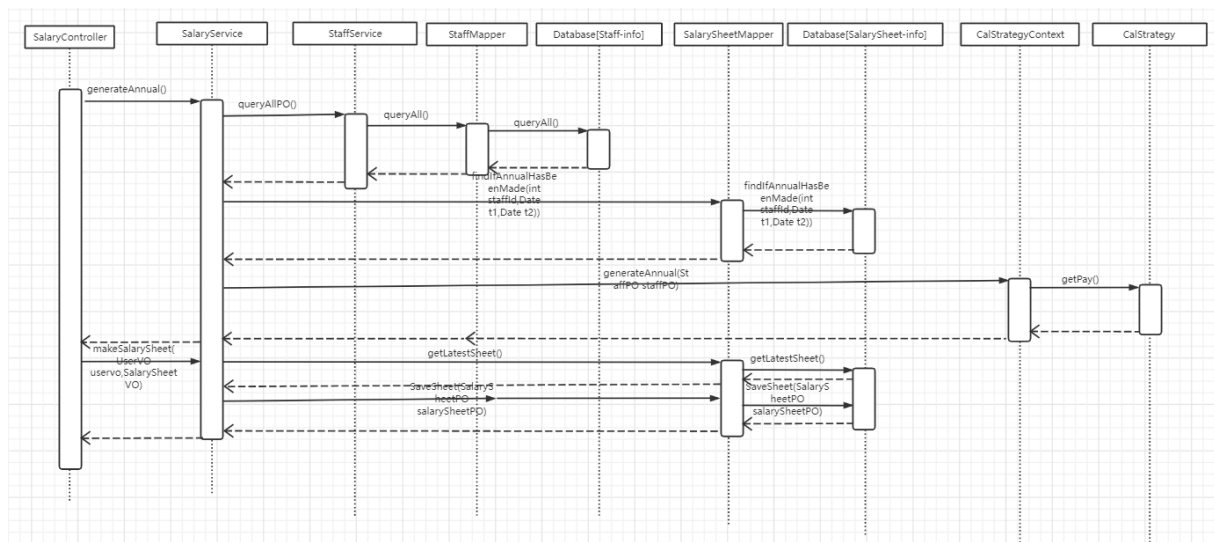
服务名	服务
SalaryService.makeSalarySheet	制定工资单
SalaryService.generateSalary	生成工资条目
SalaryService.generateAnnual	生成年终奖条目
SalaryService.getSalarySheetByState	根据审批状态获取工资单
SalaryService.approval	审批工资单
SalaryService.getSalarySheetById	根据工资单 ID 获取工资单

(4) 业务逻辑层的动态模型

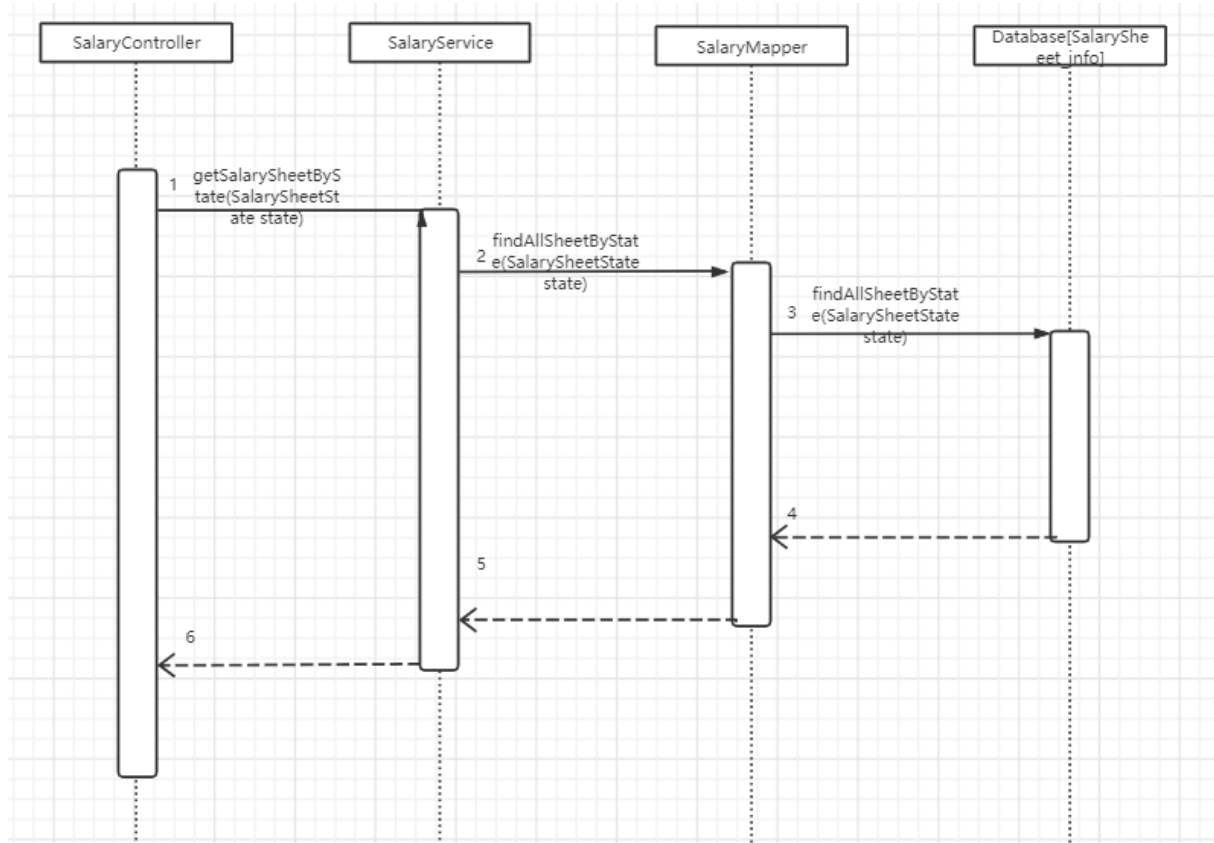
1. 下图表明了 ERP 系统中，Salary.makeSalary 的相关对象之间的协作。



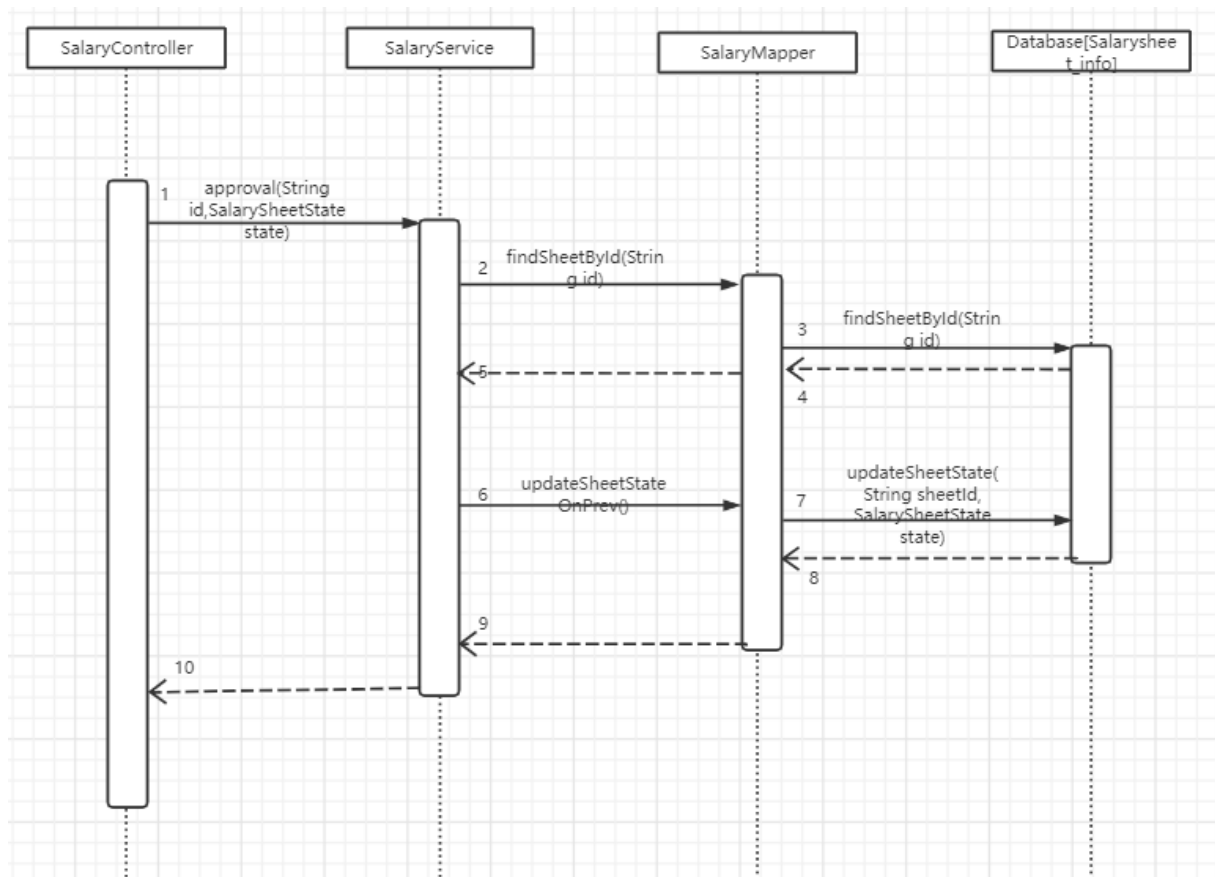
2. 下图表明了 ERP 系统中，Salary.makeAnnual 的相关对象之间的协作。



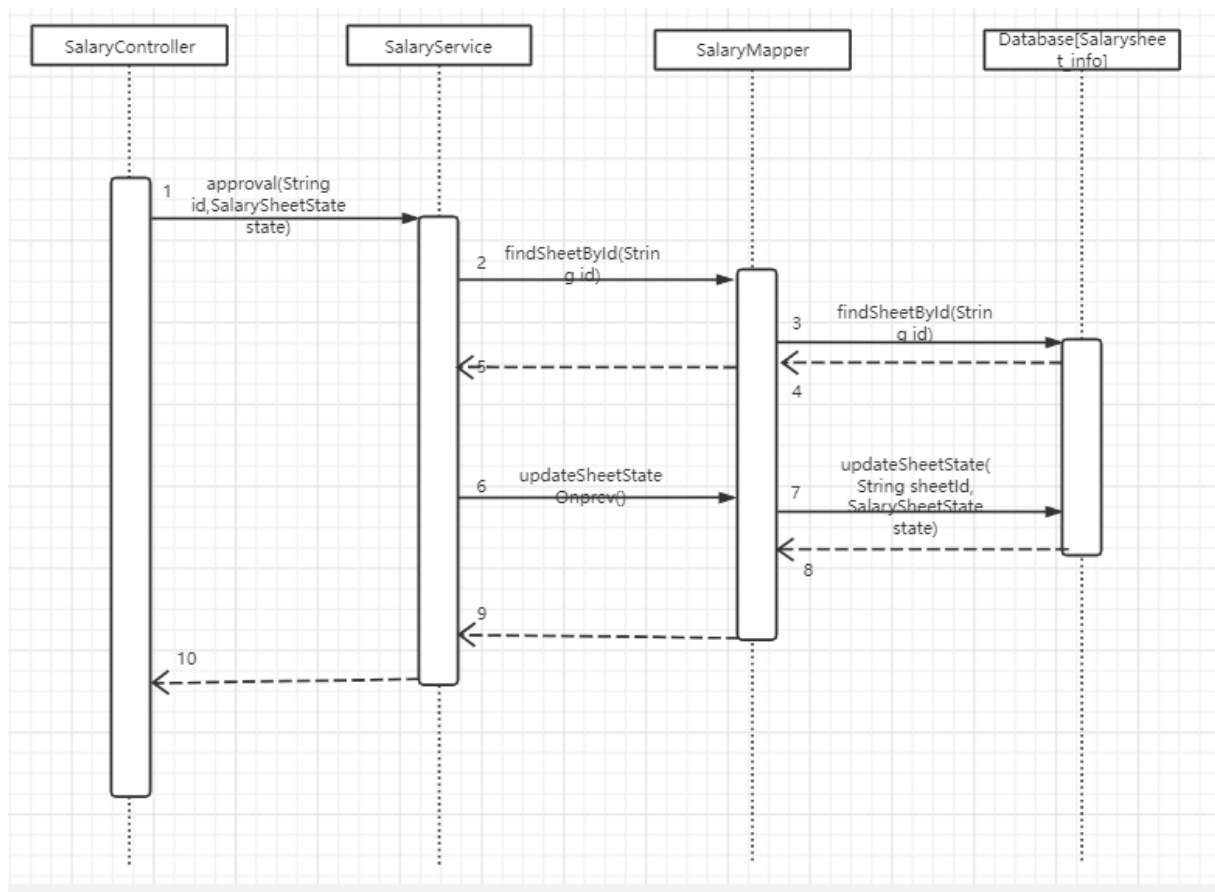
3. 下图表明了 在 ERP 系统中，Salary.showSheetByState 的相关对象之间的协作。



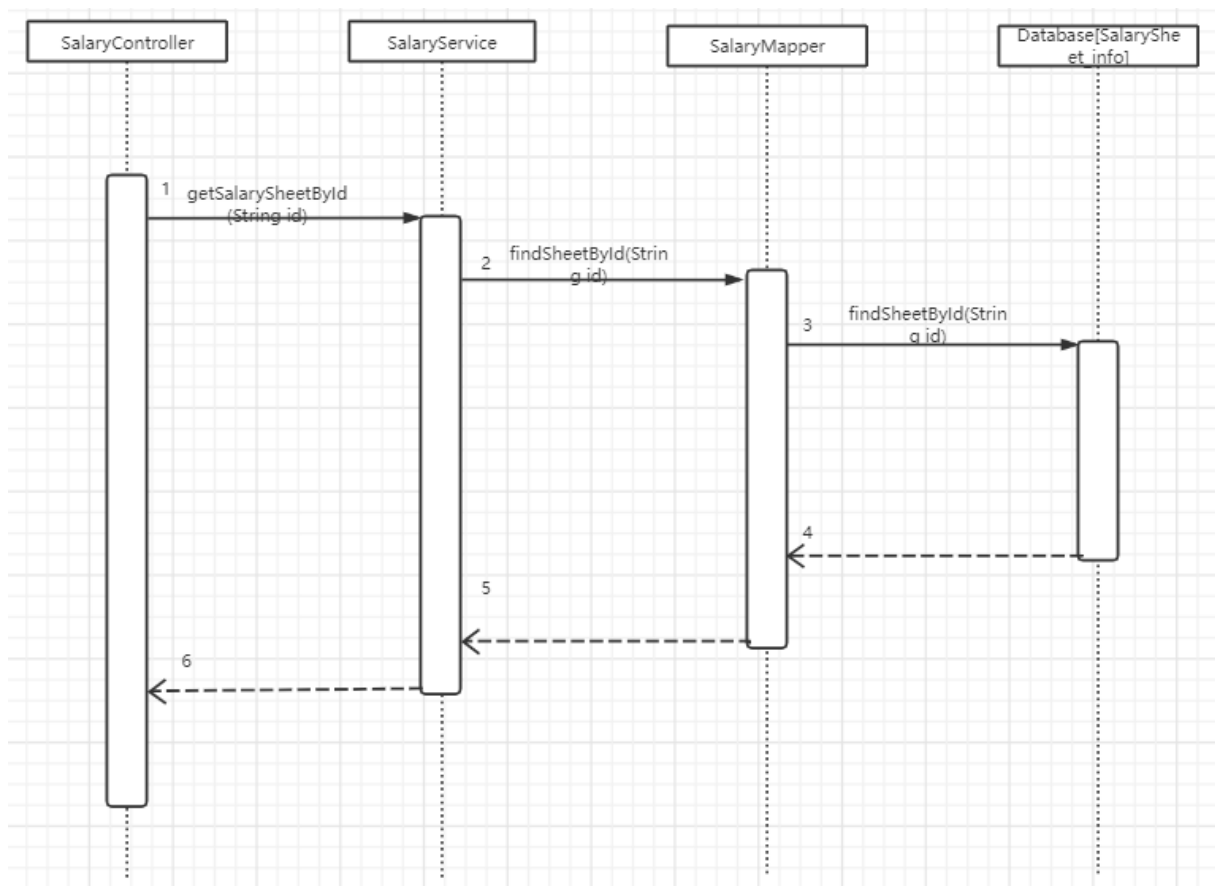
4. 下图表明了 在 ERP 系统中，Salary.firstApproval 的相关对象之间的协作。



5. 下图表明了 在 ERP 系统中，Salary.secondApproval 的相关对象之间的协作。



6. 下图表明了 在 ERP 系统中, Salary.findBySheetId 的相关对象之间的协作。



(5) 业务逻辑层的设计原理

利用委托式风格，每个界面需要访问的业务逻辑由各自的控制器委托给不同的领域对象。

4.1.5 Promotionbl 模块

(1) 模块概述

Promotionbl 模块承担的需求参见需求规格说明文档功能需求及相关非功能需求说明文档。

Promotionbl 模块的职责及接口参见软件系统结构描述文档。

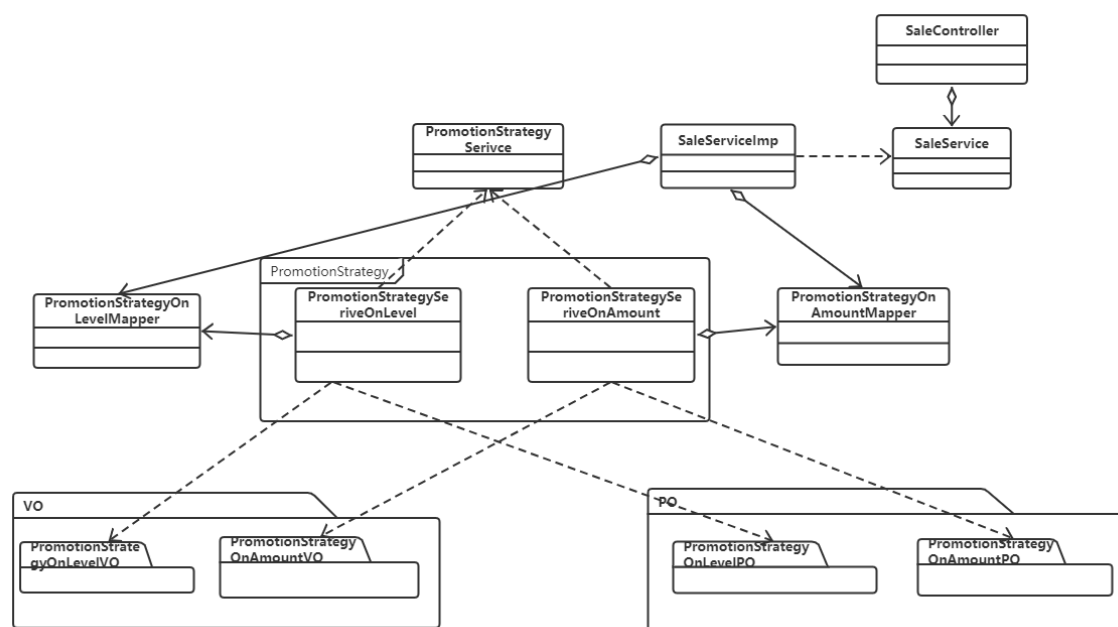
(2) 整体结构

根据体系结构的设计，我们将系统分为了前端展示层、Controller 层、业务逻辑层、数据层。

每一层为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们修改了 SaleService 接口，业务逻辑层和数据层之间设置了 Mapper 接口。

PromotionStrategyOnLevelPO 和 PromotionStrategyOnAmountPO 是作为两种促销策略（等级促销策略和总价促销策略）的持久化对象而添加到设计模型中。在该类中我们使用了策略模式，用来实现对促销策略的灵活变更。

Promotionbl 模块的设计如下图：



各个类的职责：

模块	职责
SaleController	负责实现促销策略信息前后端的数据交互
SaleServiceImp	负责实现促销策略信息的增删改查功能
PromotionStrategyOnLevelVO	等级促销策略信息增删改查的实现载体，存有编号、顾客等级、折扣等信息。
PromotionStrategyOnLevelPO	等级促销策略信息增删改查的实现载体，存有编号、顾客等级、折扣等信息。
PromotionStrategyOnAmountVO	总价促销策略信息增删改查的实现载体，存有编号、最低价最高价、折扣等信息。
PromotionStrategyOnAmountPO	总价促销策略信息增删改查的实现载体，存有编号、最低价最高价、折扣等信息。

(3) 模块内部类的接口规范

Promotionbl 模块的各个类的职责和内部类的接口规范已在 ERP 软件体系结构文档中描述。

接口名字	语法	前置条件	后置条件
SaleController.makePromotionStrategyOnLevel	makePromotionStrategyOnLevel(PromotionStrategyOnLevelVO promotionStrategyOnLevelVO)	无	将促销策略信息传递给 promotionStrategyOnLevel
SaleController.getAllPromotionStrategyOnLevel	getAllPromotionStrategyOnLevel()	无	从 promotionStrategyOnLevel 获取促销策略信息
SaleController.updatePromotionStrategyOnLevel	updatePromotionStrategyOnLevel(PromotionStrategyOnLevelVO promotionStrategyOnLevelVO)	需要编辑的策略已被建立	将促销策略信息传递给 promotionStrategyOnLevel
SaleController.deletePromotionStrategyOnLevel	deletePromotionStrategyOnLevel(Integer id)	需要删除的策略已被建立	从 promotionStrategyOnLevel 删除促销策略信息
SaleController.makePromotionStrategyOnAmount	makePromotionStrategyOnAmount(PromotionStrategyOnAmountVO promotionStrategyOnLevelVO)	无	将促销策略信息传递给 promotionStrategyOnAmount
SaleController.getAllPromotionStrategyOnAmount	getAllPromotionStrategyOnAmount()	无	从 promotionStrategyOnAmount 获取促销策略信息

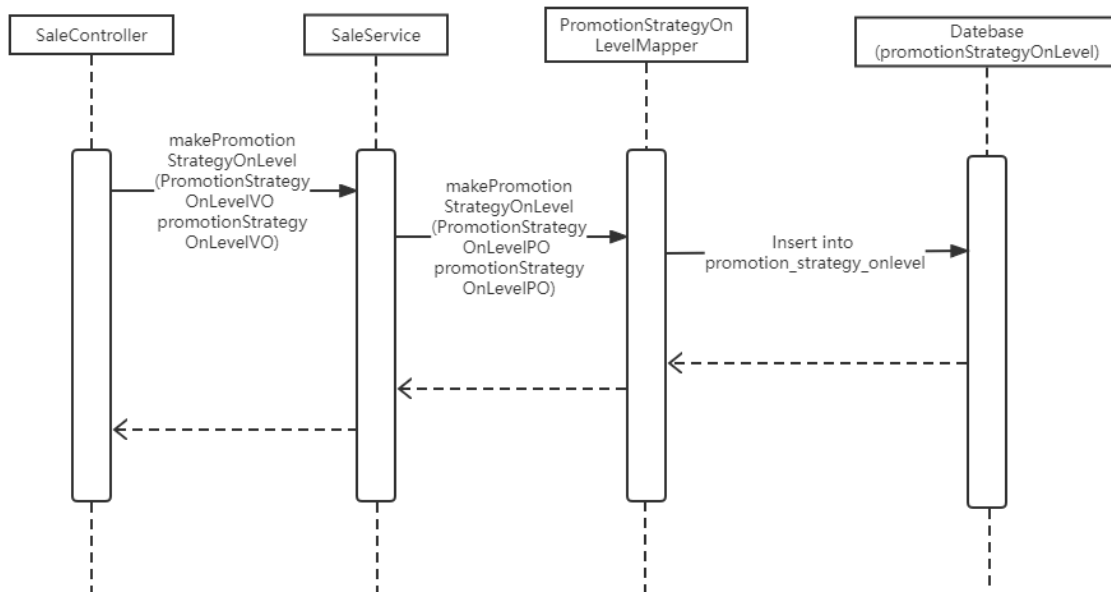
SaleController.updatePromotionStrategyOnAmount	updatePromotionStrategyOnAmount(PromotionStrategyOnAmountVO promotionStrategyOnAmountVO)	需要编辑的策略已被建立	将促销策略信息传递给 promotionStrategyOnAmount
SaleController.deletePromotionStrategyOnAmount	deletePromotionStrategyOnAmount(Integer id)	需要删除的策略已被建立	从 promotionStrategyOnAmount 删除促销策略信息

Promotion 模块的服务接口

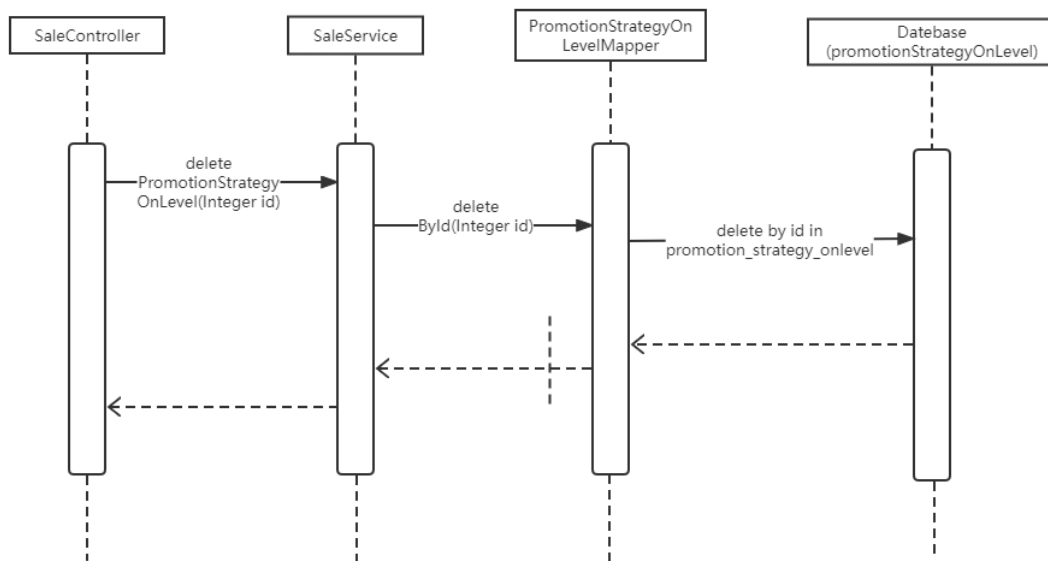
服务名	服务
SaleService.makePromotionStrategyOnLevel	制定关于等级的促销策略
SaleService.getAllPromotionStrategyOnLevel	获取所有的关于等级的促销策略
SaleService.updatePromotionStrategyOnLevel	更新关于等级的促销策略
SaleService.deletePromotionStrategyOnLevel	删除对应的等级促销策略
SaleService.makePromotionStrategyOnAmount	制定关于总价的促销策略
SaleService.getAllPromotionStrategyOnAmount	获取所有的关于总价的促销策略
SaleService.updatePromotionStrategyOnAmount	更新关于总价的促销策略
SaleService.deletePromotionStrategyOnAmount	删除对应的总价促销策略

(4) 业务逻辑层的动态模型

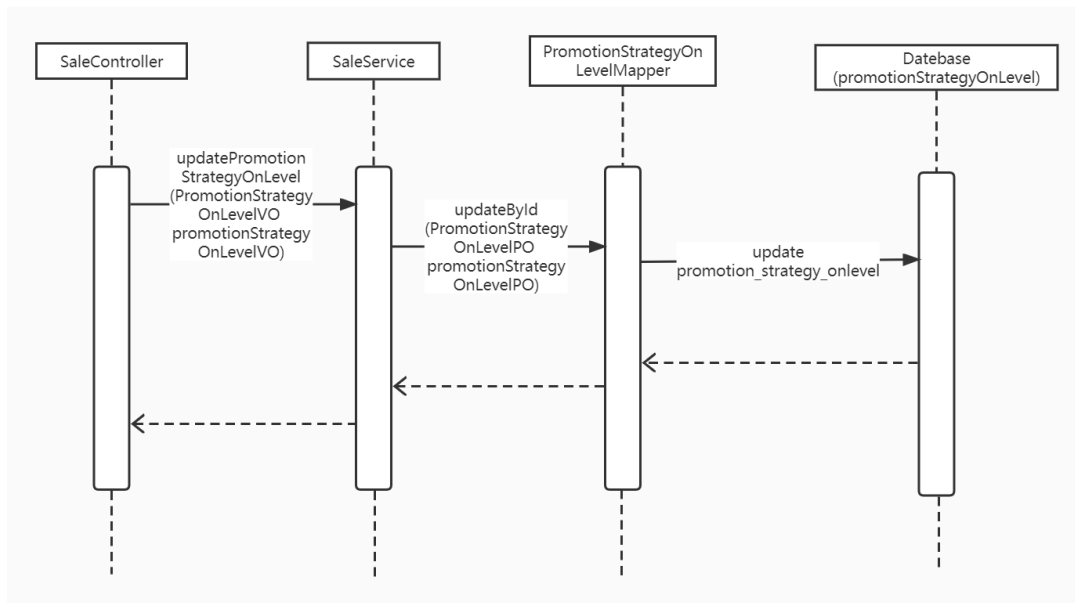
1. 下图表明了 在 ERP 系统中，Sale.makePromotionStrategyOnLevel 的相关对象之间的协作。



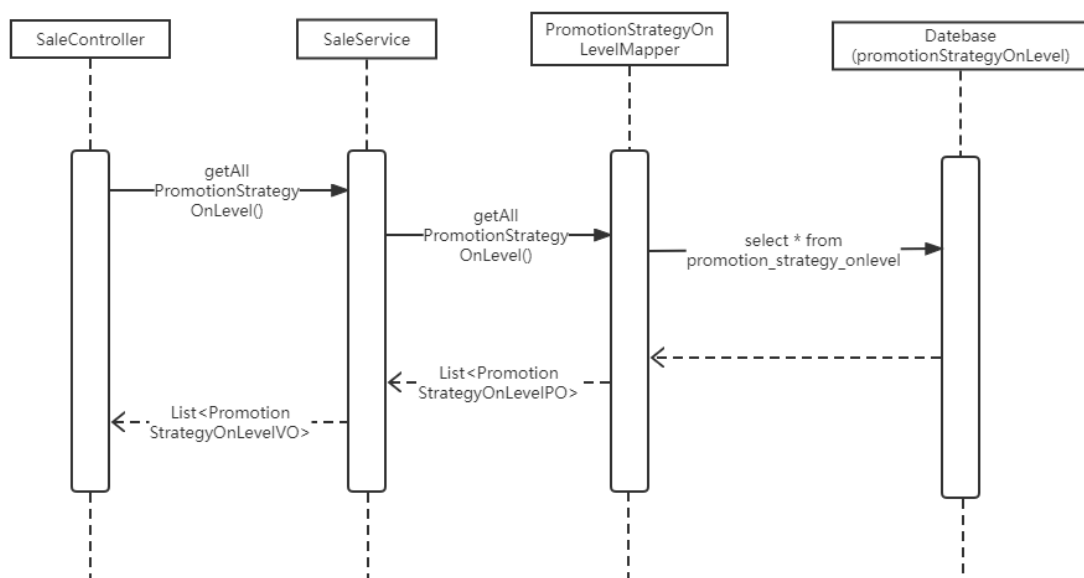
2. 下图表明了 在 ERP 系统中，Sale.deletePromotionStrategyOnLevel 的相关对象之间的协作。



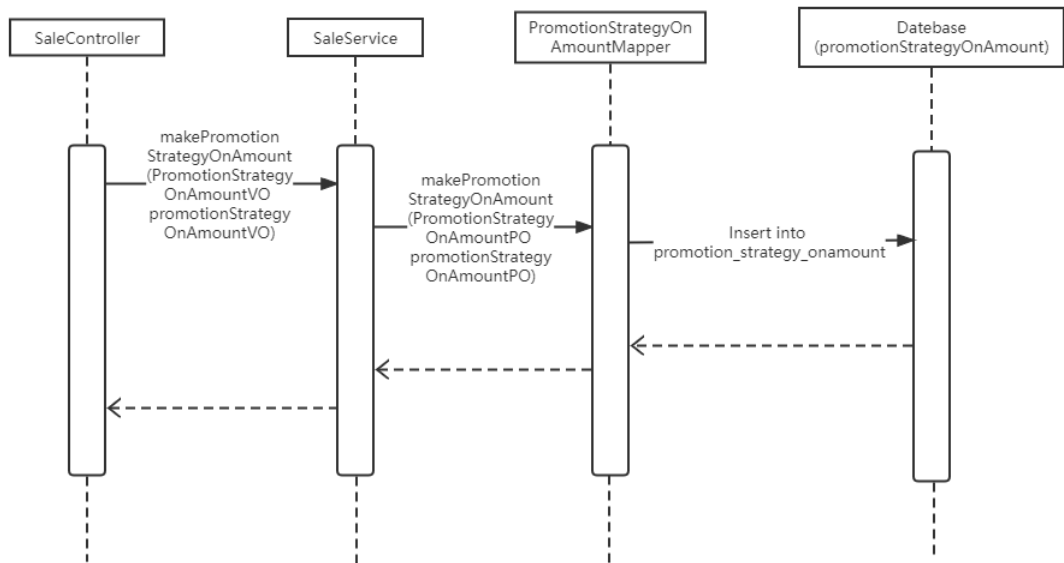
3. 下图表明了 在 ERP 系统中，Sale.updatePromotionStrategyOnLevel 的相关对象之间的协作。



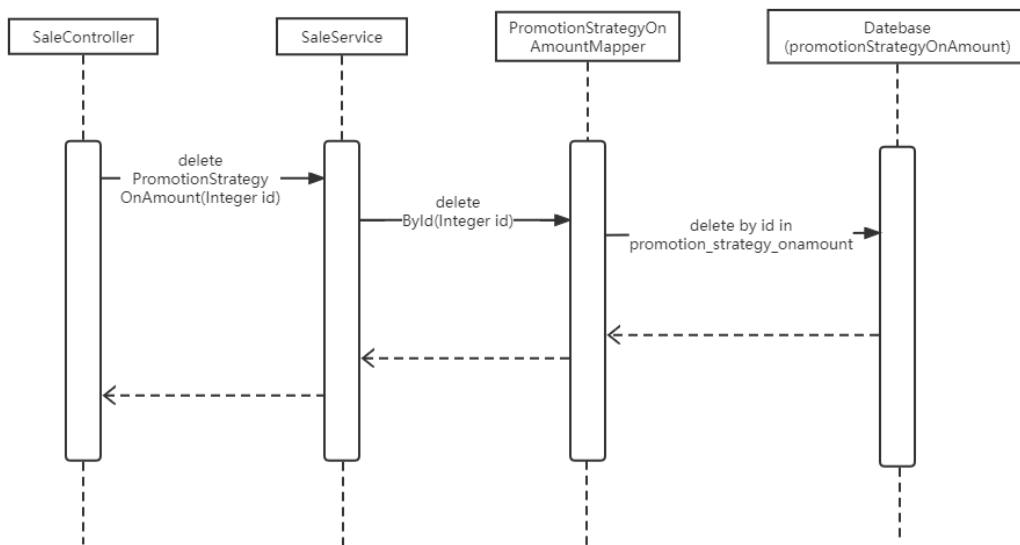
4. 下图表明了 在 ERP 系统中，Sale.getAllPromotionStrategyOnLevel 的相关对象之间的协作。



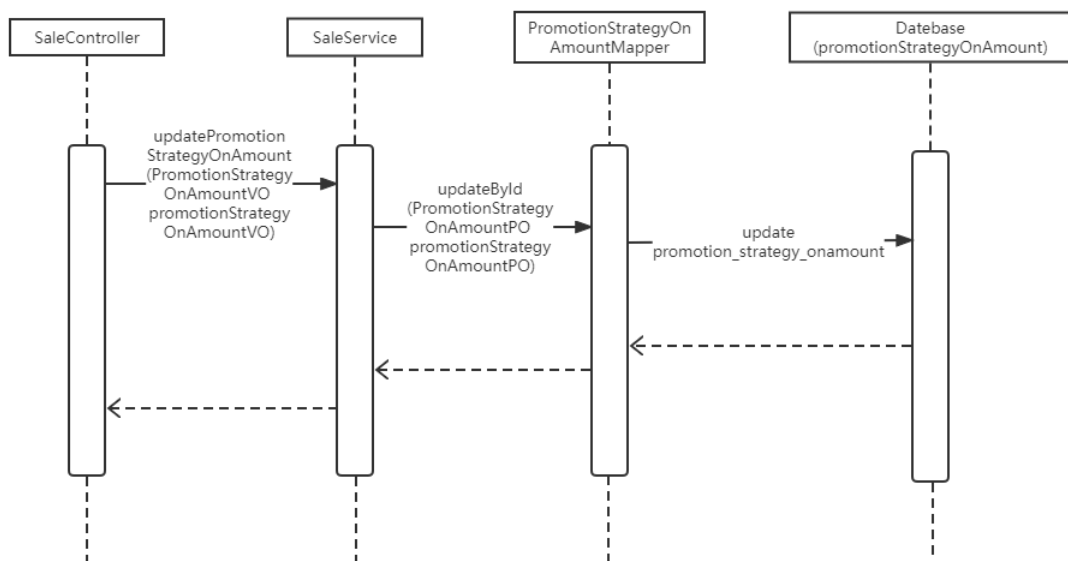
5. 下图表明了 在 ERP 系统中，Sale.makePromotionStrategyOnAmount 的相关对象之间的协作。



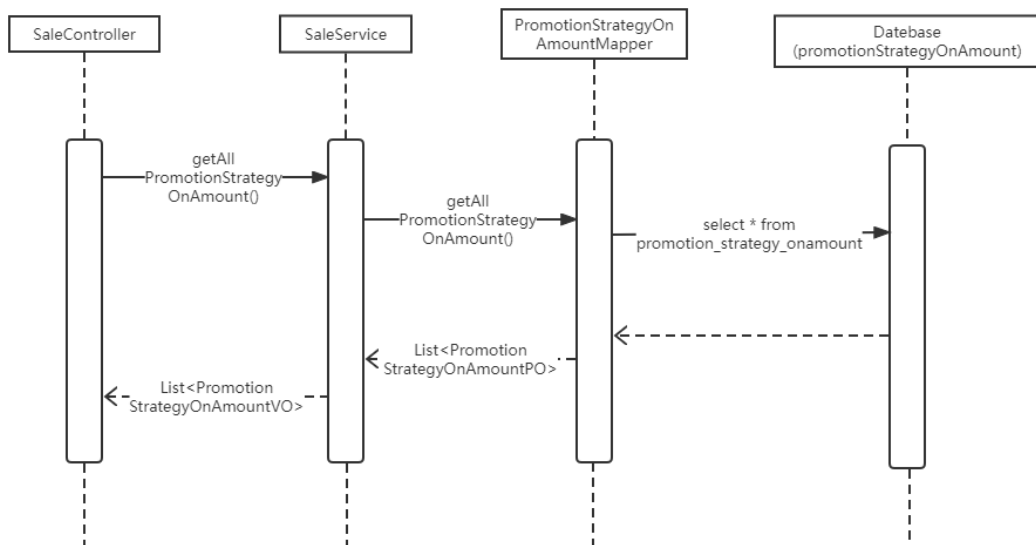
6. 下图表明了 在 ERP 系统中，Sale.deletePromotionStrategyOnAmount 的相关对象之间的协作。



7. 下图表明了 在 ERP 系统中，Sale.updatePromotionStrategyOnAmount 的相关对象之间的协作。



8. 下图表明了 在 ERP 系统中，Sale.getAllPromotionStrategyOnAmount 的相关对象之间的协作。



(5) 业务逻辑层的设计原理

利用委托式控制风格，每个界面需要访问的业务逻辑由各自的控制器委托给不同的领域对象。

5. 依赖视角

下图分别是前端和后端各自的包之间的依赖关系

