(1)模块概述

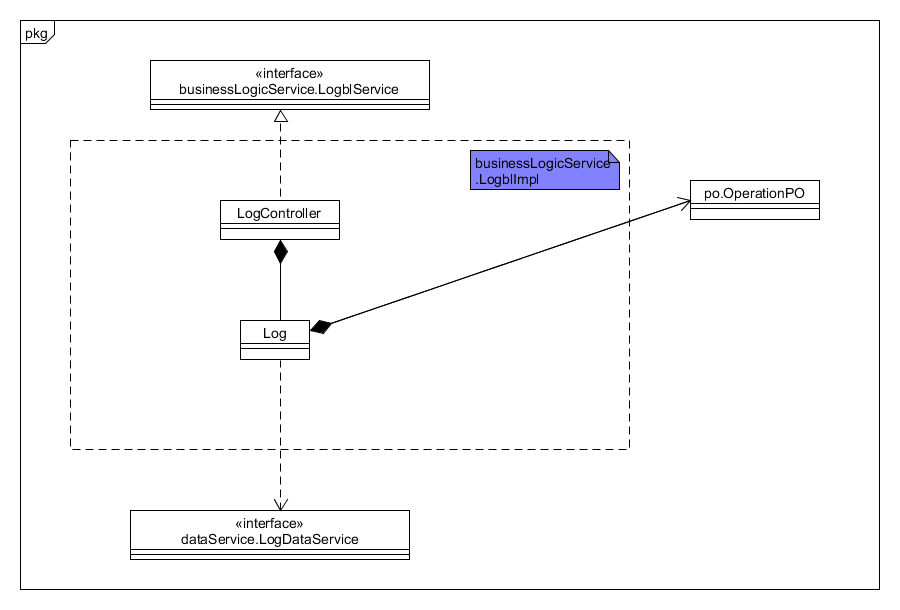
Logbl模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

Logbl模块的职责和接口参见软件体系结构描述文档。

(2)整体结构

根据体系结构的设计，系统分为展示层、业务逻辑层和数据层，并在展示层和业务逻辑层之间添加bussinssLogicServi ce.LogblService接口、在业务逻辑层和数据层之间添加dataService.LogDataService接口，从而增加了系统的灵活性。我们增加了LogController从而隔离了业务逻辑职责和逻辑控制职责，这样LogController会将业务逻辑处理委托给Logbl对象。LogPO作为用户信息的持久化对象被添加到设计模型中去的。LogImpl用来进行对管理用户操作时产生的系统日志的记录。

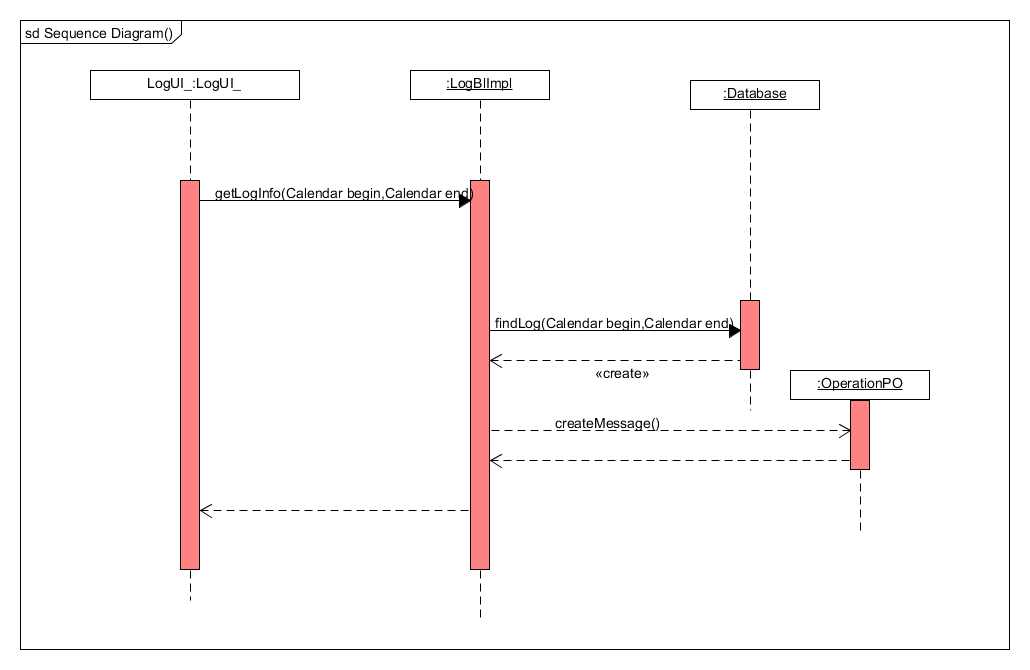
Logbl模块的设计如图所示。



Logbl模块的各个类的职责如下表所示。

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Log.getLogInfo | 语法 | public Arraylist<OperationVO> getLogInfo(Calendar begin, Calendar end) |
| 前置条件 | Begin的时间在end之前 |
| 后置条件 | 查找该时间段内的操作记录并返回 |
| Log.addLog | 语法 | public Boolean addLog(OperationVO op) |
| 前置条件 | 无 |
| 后置条件 | 如果增加成功返回true，否则返回false |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| LogDataService  .addLog(OperationPo operation):boolean | 增加传入的操作单一持久化对象 | |
| LogDataService  .findLog(Calendar begin, Calendar end):List<OperationPo> | 查找传入的时间段内的操作持久化对象并返回 | |

(4)业务逻辑层的动态模型

下图表示了物流管理系统中，当用户选择查询系统日志时，Logbl处理的相关对象的协作。

(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。