**4.2.5repertorybl模块**

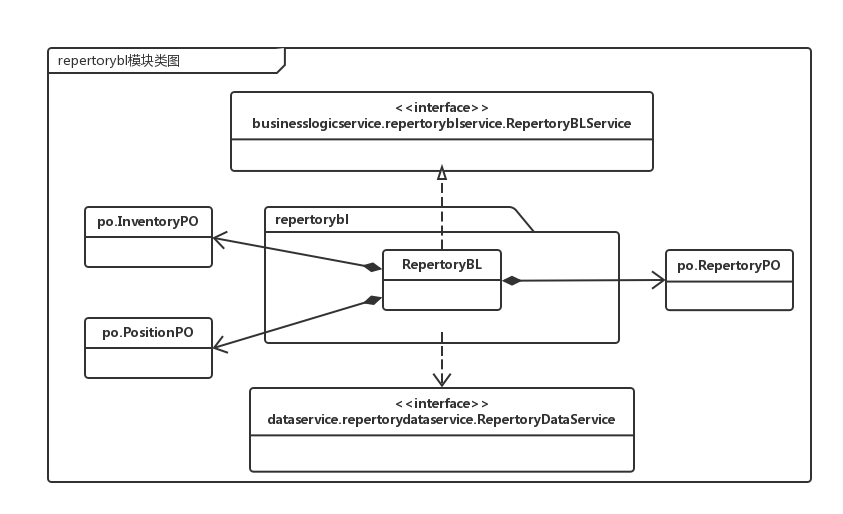
**4.2.5.1模块概述**

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

repertorybl模块的职责及接口参见ELS体系结构设计文档表13。

**4.2.5.2整体结构**

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们添加businesslogicservice.repertoryblservice.RepertoryBLService接口。业务逻辑层和数据层之间添加 dataservice.repertorydataservice.RepertoryDataService接口。因为业务逻辑处理较少，根据专家模式没有添加Controller类。RepertoryPO等PO对象是作为仓库管理员管理记录的持久化对象被添加到设计模型中去的。

repertorybl模块的设计如图5-1所示。

**图5-1 repertorybl模块各个类的设计**

repertorybl模块各个类的职责如表5-1所示。

**表5-1 repertorybl模块各个类的职责**

|  |  |
| --- | --- |
| 模块 | 职责 |
| RepertoryBL | 仓库的领域模型对象，拥有仓库数据的编号、所属管理员编号、最大容量、警戒比例，可以解决仓库管理员仓库管理的问题。 |

**4.2.5.3模块内部类的接口规范**

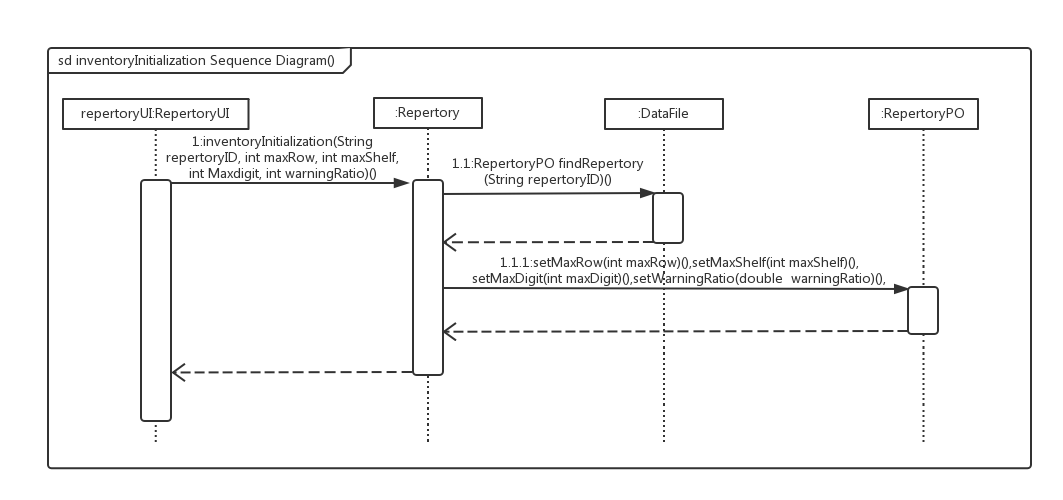
RepertoryBL的接口规范如表5-2所示。

**表5-2 OrganizationBL模块的接口规范**

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| Repertory.inventoryInitialization | 语法 | public int inventoryInitialization(String repertoryID, int maxRow, int maxShelf, int Maxdigit, int warningRatio) | |
| 前置条件 | 经过验证的仓库管理员有库存信息初始化的权利，且仓库管理员输入了有效的库存信息值 | |
| 后置条件 | 系统保存仓库初始化信息 | |
| Repertory.enterRepertory | 语法 | public int enterRepertory(String repertoryID, String JJD-ID, int blockNum, int rowNum, int shelfNum, int digitNum, String date) | |
| 前置条件 | 经过验证的仓库管理员有将商品入库的权利，且仓库管理员输入了有效的商品信息和对应仓库位置、时间信息 | |
| 后置条件 | 系统保存商品入库的位置和时间信息，生成入库单 | |
| Repertory.leaveRepertory | 语法 | public int leaveRepertory(String repertoryID, String JJD-ID, int transType, String date) | |
| 前置条件 | 经过验证的仓库管理员有将商品出库的权利，且仓库管理员输入了有效的商品信息和转运方式、时间信息 | |
| 后置条件 | 系统保存商品出库的转运方式和时间，生成出库单 | |
| Repertory.inventoryWarning | 语法 | public boolean inventoryWarning() | |
| 前置条件 | 当某区的库存总量与某区最大容量的比值达到或者超过警戒比例时，系统提示仓库管理员该区库存已达警戒比例 | |
| 后置条件 | 无 | |
| Repertory.inventoryCheck | 语法 | public ArrayList<InventoryVO> inventoryCheck(String repertoryID, String beginDate, String endDate) | |
| 前置条件 | 经过验证的仓库管理员有库存查看的权利，且仓库管理员输入了有效的时间段信息 | |
| 后置条件 | 系统根据输入的时间段查找对应的出入库商品，并返回这段时间出/入库商品数量和金额总和，库存数量合计 | |
| Repertory.inventoryStockTaking | 语法 | public ArrayList<InventoryVO> inventoryStockTaking(String repertoryID) | |
| 前置条件 | 经过验证的仓库管理员有库存盘点的权利，且仓库管理员输入了库存盘点命令 | |
| 后置条件 | 系统自动根据提交库存盘点命令的时间生成一个截止点，并根据该截止点查找当天到截止点之前的仍在仓库中的商品，返回商品信息 | |
| 需要的服务（需接口） | | | |
| 服务名 | | | 服务 |
| FileFactory.getRepertoryData | | | 得到Repertory文件的引用 |
| RepertoryDataService.modifyRepertory(RepertoryPO repertorypo) | | | 在仓库数据文件中插入一个RepertoryPO对象 |
| RepertoryDataService.findRepertory(String repertoryID) | | | 在仓库数据文件中删除一个RepertoryPO对象 |
| RepertoryDataService.addGoods(String repertoryID, GoodsPO goodspo) | | | 在编号为repertoryID的仓库对应库存列表中插入一个GoodsPO对象 |
| RepertoryDataService.deleteGoods(String repertoryID, GoodsPO goodspo) | | | 在编号为repertoryID的仓库对应库存列表中删除一个GoodsPO对象 |
| RepertoryDataService.modifyGoods(String repertoryID, GoodsPO goodspo) | | | 在编号为repertoryID的仓库对应库存列表中更新一个GoodsPO对象 |
| RepertoryDataService.findGoodsbyID(String repertoryID, String JJD-ID) | | | 根据输入的仓库编号和订单号，返回对应的GoodsPO对象 |
| RepertoryDataService.findGoodsbyDate(String repertoryID, String beginDate, String endDate) | | | 根据输入的仓库编号和起始日期，返回对应的GoodsPO对象 |
| RepertoryDataService.findGoodsbyTime(String repertoryID, String time) | | | 根据输入的仓库编号和截止点时间，返回对应的GoodsPO对象 |

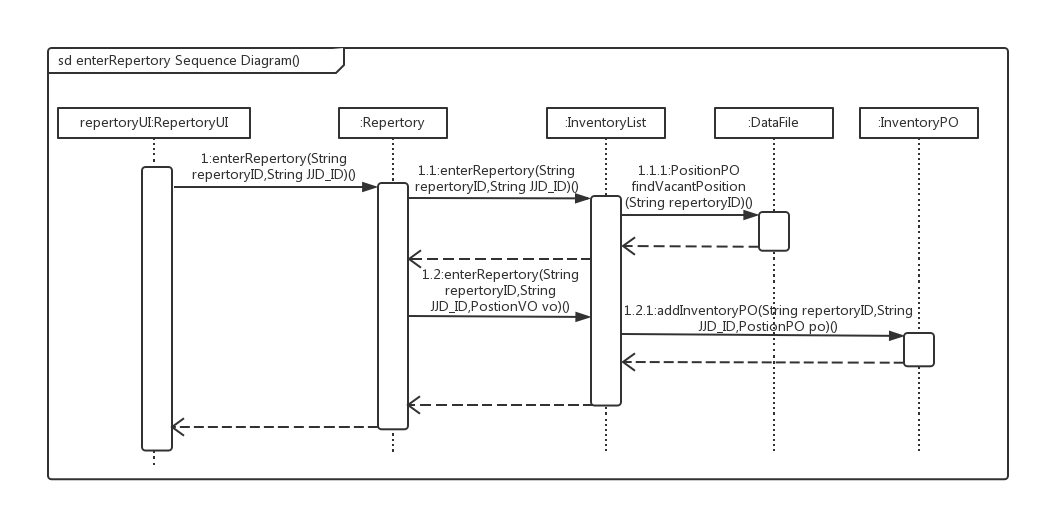
**4.5.2.4 业务逻辑层的动态模型**

图5-2表明了快递物流系统中，当仓库管理员开始库存信息初始化任务后，仓库管理业务逻辑处理的相关对象的协作。



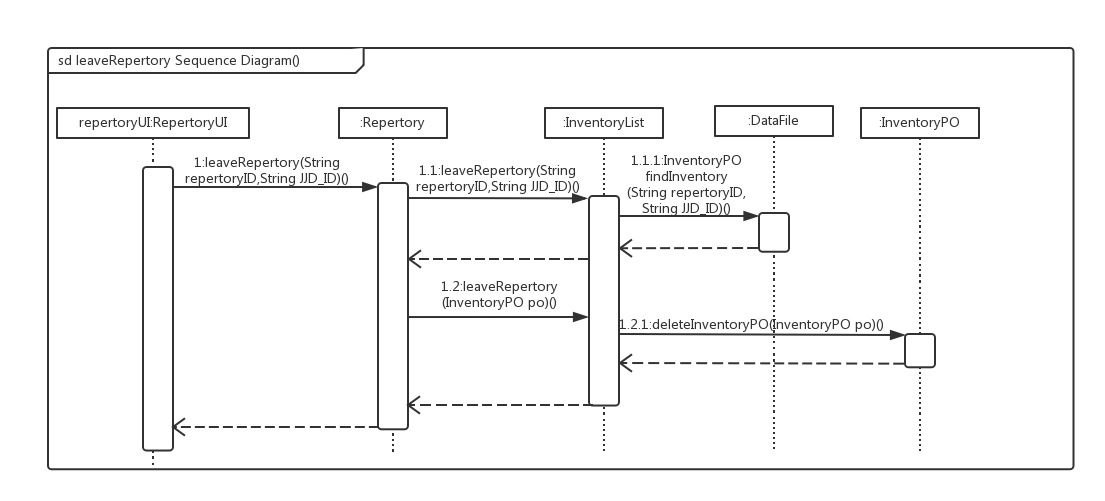
**图5-2 库存信息初始化的顺序图**

图5-3表明了快递物流系统中，当仓库管理员开始入库任务后，仓库管理业务逻辑处理的相关对象的协作。



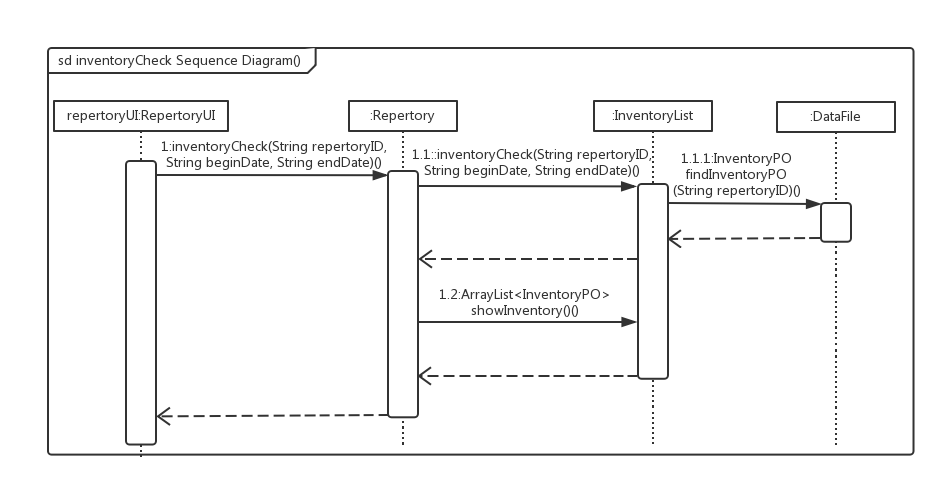
**图5-3 入库的顺序图**

图5-4表明了快递物流系统中，当仓库管理员开始出库任务后，仓库管理业务逻辑处理的相关对象的协作。



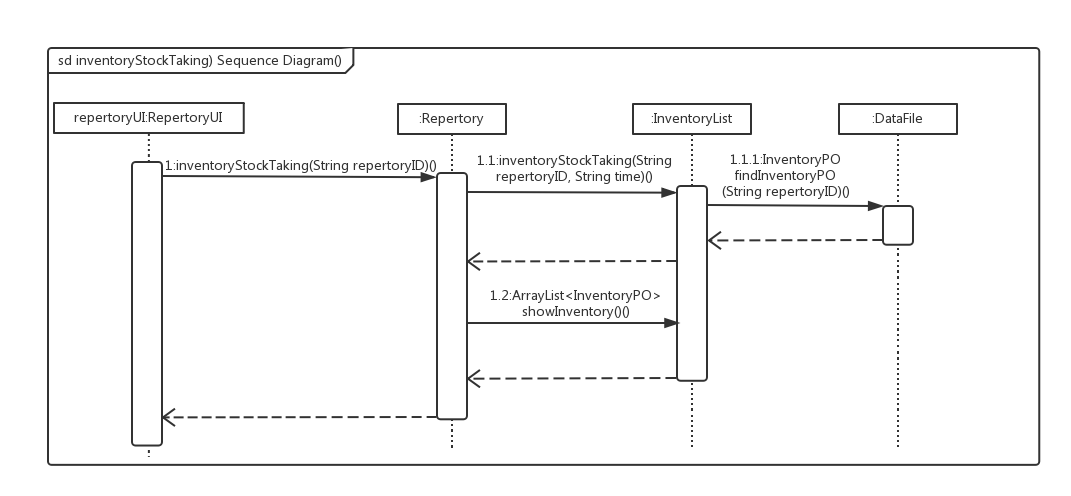
**图5-4 出库的顺序图**

图5-5表明了快递物流系统中，当仓库管理员开始库存查看任务后，仓库管理业务逻辑处理的相关对象的协作。



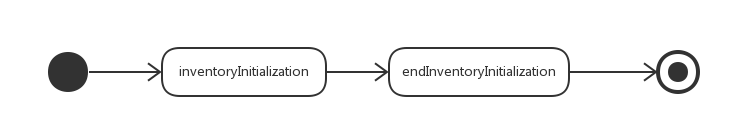
**图5-5 库存查看的顺序图**

图5-6表明了快递物流系统中，当仓库管理员开始库存盘点任务后，仓库管理业务逻辑处理的相关对象的协作。

****

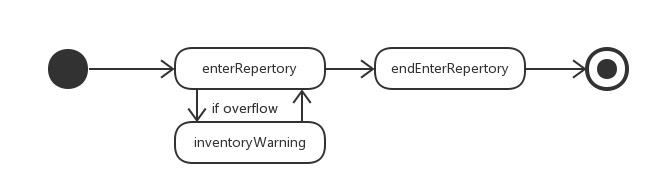
**图5-6 库存盘点的顺序图**

如图5-7所示的状态图描述了RepertoryBL对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inventoryInitialization方法被UI调用，RepertoryBL进入inventoryInitialization状态；仓库管理员输入最大容量和警戒比例后，结束库存信息初始化进入endInventoryInitialization状态。



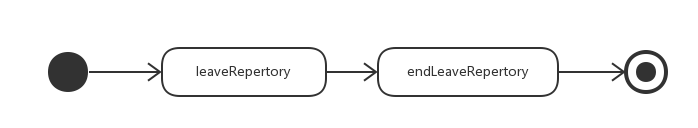
**图5-7 库存信息初始化的状态图**

如图5-8所示的状态图描述了RepertoryBL对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着enterRepertory方法被UI调用，RepertoryBL进入enterRepertory状态；当仓库管理员选择分区后，系统根据被选择分区是否超过警戒比例进入inventoryWarning状态要求再次确认入库，或者endEnterRepertory状态完成入库操作。



**图5-8 入库的状态图**

如图5-9所示的状态图描述了RepertoryBL对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着leaveRepertory方法被UI调用，RepertoryBL进入leaveRepertory状态；仓库管理员确认后，最后进入endEnterRepertory状态完成出库操作。



**图5-9 出库的状态图**

如图5-10所示的状态图描述了RepertoryBL对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inventoryCheck方法被UI调用，RepertoryBL进入inventoryCheck状态；最后通过endInventoryCheck方法确定查看结束，进入endShow状态。



**图5-10 库存查看的状态图**

如图5-11所示的状态图描述了RepertoryBL对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着inventoryStockTaking方法被UI调用，RepertoryBL进入inventoryStockTaking状态；最后通过endInventoryStockTaking方法确定查看结束，进入endShow状态。

****

**图5-11 库存盘点的状态图**