5.3 业务逻辑层的分解

业务逻辑层包括多个针对界面的业务逻辑处理对象。例如，User对象负责处理用户管理和登录界面的业务逻辑，OrderNew对象负责新建单据界面的业务逻辑。业务逻辑层的设计如图所示。

5.3.1 业务逻辑层模块的职责

业务逻辑层模块的职责如表所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| userLogic | 负责实现用户管理和登录界面所需要的服务 |
| transitInfoLogic | 负责实现物流信息查询界面所需要的服务 |
| orderNewLogic | 负责实现单据生成界面所需要的服务 |
| orderApproveLogic | 负责实现单据审批界面所需要的服务 |
| accountLogic | 负责实现账户管理界面所需要的服务 |
| cityLogic | 负责实现城市管理界面所需要的服务 |
| workOrgManLogic | 负责实现人员机构管理界面所需要的服务 |
| stockLogic | 负责实现库存管理界面所需要的服务 |
| checkLogic | 负责实现统计报表与查看日志界面所需要的服务 |

5.3.2 业务逻辑层模块的接口规范

各逻辑层模块接口规范如后表所示。

userLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| User.login | 语法 | public ResultMessage login(String id,String password) |
| 前置条件 | id和password均输入 |
| 后置条件 | 查找是否存在相应的用户，根据输入的id和password返回结果和用户信息 |
| User.newUser | 语法 | public boolean newUser(UserVO user) |
| 前置条件 | 输入所有用户信息后，确认新建 |
| 后置条件 | 将新用户信息加入数据库，返回加入结果 |
| User.deleteUser | 语法 | public boolean deleteUser(String id) |
| 前置条件 | 选择了一个用户，选择删除 |
| 后置条件 | 在数据库中除去相应用户的信息，返回删除结果 |
| User.reviseUser | 语法 | public boolean reviseUser(UserVO user) |
| 前置条件 | 修改了用户的信息，确认 |
| 后置条件 | 在数据库中修改相应用户的信息，返回修改结果 |
| User.checkUser | 语法 | public ResultMessage checkUser(String id) |
| 前置条件 | 选择了一个用户，选择查看 |
| 后置条件 | 数据库中查找相应用户信息，返回信息和查看结果 |
| User.userList | 语法 | public ResultMessage userList() |
| 前置条件 | 用户选择查看用户列表 |
| 后置条件 | 返回数据库中所有用户的列表，包含用户的账号、姓名和所属机构 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| DatabaseFactory.getUserData() | 得到User数据库的服务的引用 | |
| UserDataSer.addUser(UserPO po) | 在数据库中添加UserPO对象 | |
| UserDataSer.deleteUser(String id) | 在数据库中删除id所指向的对象 | |
| UserDataSer.reviseUser(UserPO po) | 在数据库中修改UserPO中id指向的对象的信息 | |
| UserDataSer.findUser(String id) | 在数据库中找到相应用户的信息 | |
| UserDataSer.list() | 在数据库中获得用户列表 | |

transitInfo模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| TransitInfo.enterBarcodeCustom | 语法 | Public ResultMessage enterBarcodeCustom(String barcode); |
| 前置条件 | barcode符合输入规则 |
| 后置条件 | 查找是否存在相应的订单，据输入的barcode返回相应订单的物流信息和货运状态 |
| TransitInfo.enterBarcodeCourier | 语法 | Public ResultMessage enterBarcodeCourier(String barcode); |
| 前置条件 | barcode符合输入规则 |
| 后置条件 | 查找是否存在相应的订单，根据输入的barcode返回该订单所有信息 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| transitInfoDataSer.find(String  barcode) | 根据barcode进行查找订单对象 | |
| transitInfoDataSer.findTransit(String  barcode) | 根据barcode查找相应订单的物流信息和货运状态 | |
| databaseFactory.getOrdersDatabase | 得到orders数据库的服务的引用 | |

orderNewLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| OrderNew.Create | 语法 | public Order Create(Order order) |
| 前置条件 | 单据的各项信息已输入并确认 |
| 后置条件 | 将该新增单据上传至数据库，返回上传状态 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| DatabaseFactory.getOrderData | 得到Order数据库的服务的引用 | |
| orderNewDataSer.add(OrderPO po) | 在数据库中添加OrderPO对象 | |

workOrgManLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| WorkOrgMan.addOrg | 语法 | public boolean add(OrgVO org); |
| 前置条件 | 用户输入了新机构的信息 |
| 后置条件 | 将新增的机构信息加入数据库，返回加入结果 |
| WorkOrgMan.deleteOrg | 语法 | public boolean deleteOrg(String id) |
| 前置条件 | 用户选择一个机构进行删除 |
| 后置条件 | 根据机构名在数据库中删除该账户信息，返回删除结果 |
| WorkOrgMan.reviseOrg | 语法 | public boolean reviseOrg(OrgVO org) |
| 前置条件 | 修改了机构信息并确认 |
| 后置条件 | 更新数据库中该机构的信息，返回更新结果 |
| WorkOrgMan. orgList | 语法 | public ResultMessage orgList() |
| 前置条件 | 用户选择查看机构列表 |
| 后置条件 | 返回数据库中所有机构的列表 |
| WorkOrgMan.checkOrg | 语法 | public ResultMessage checkOrg(String id) |
| 前置条件 | 用户选择一个机构进行查看 |
| 后置条件 | 返回该机构信息 |
| WorkOrgMan.searchOrg | 语法 | public ResultMessage searchOrg(String keywords) |
| 前置条件 | 用户检索机构并输入关键字 |
| 后置条件 | 根据该关键字查找符合条件的机构并返回 |
| WorkOrgMan.addWork | 语法 | public ResultMessage addWork(WorkVO work); |
| 前置条件 | 用户输入了新人员信息 |
| 后置条件 | 将新增的人员信息加入数据库，返回加入结果 |
| WorkOrgMan.deleteWork | 语法 | public ResultMessage deleteWork(String id) |
| 前置条件 | 用户选择一个人员进行删除 |
| 后置条件 | 根据人员名在数据库中删除该人员信息，返回删除结果 |
| WorkOrgMan.reviseWork | 语法 | public ResultMessage reviseWork(WorkVO work) |
| 前置条件 | 修改了人员信息并确认 |
| 后置条件 | 更新数据库中该人员的信息，返回更新结果 |
| WorkOrgMan. workList | 语法 | public ResultMessage workList() |
| 前置条件 | 用户选择查看人员列表 |
| 后置条件 | 返回数据库中所有人员的列表 |
| WorkOrgMan.checkWork | 语法 | public ResultMessage checkWork(String id) |
| 前置条件 | 用户选择一个人员进行查看 |
| 后置条件 | 返回该人员信息 |
| WorkOrgMan.searchWork | 语法 | public ResultMessage searchWork(String keywords) |
| 前置条件 | 用户检索人员并输入关键字 |
| 后置条件 | 根据该关键字查找符合条件的人员并返回 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| workOrgManDataSer.findWork(String keywords) | 从数据库中找到含有该关键字的人员，返回数据 | |
| workOrgManDataSer.addWork(WorkPO  po) | 在数据库中新增一个人员的数据 | |
| workOrgManDataSer.deleteWork(String id) | 在数据库中删除一个人员的数据 | |
| workOrgManDataSer.reviseWork(WorkPO po) | 在数据库中修改一个人员的数据 | |
| workOrgManDataSer.checkWork(String id) | 在数据库中获得一个人员的数据，返回 | |
| DatabaseFactory.getWorkbase() | 得到Work数据库的服务的引用 | |
| workOrgManDataSer.findOrg(String keywords) | 在数据库中找到含有该关键字的机构，返回数据 | |
| workOrgManDataSer.addOrg(OrgPO  po) | 在数据库中新增一个机构的信息 | |
| workOrgManDataSer.deleteOrg(String id  ) | 在数据库中删除一个机构的信息 | |
| workOrgManDataSer.reviseOrg(OrgPO po) | 在数据库中修改一个机构的信息 | |
| workOrgManDataSer.checkOrg(String id) | 在数据库中获得一个机构的信息，返回数据 | |
| DatabaseFactory.getOrgbase | 得到org数据库的服务的引用 | |

cityLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| City.addCity | 语法 | public ResultMessage addCity(String name); |
| 前置条件 | 增加一个城市 |
| 后置条件 | 将该城市信息加入数据库，返回添加结果 |
| City.enterDistance | 语法 | public ResultMessage enterDistance(long distance); |
| 前置条件 | 总经理已选定了两个城市，distance符合输入规则 |
| 后置条件 | 将两城市距离信息加入数据库，或更新数据库已有信息，返回添加结果 |
| City.enterPrice | 语法 | public ResultMessage enterPrice(double price); |
| 前置条件 | 总经理已经选定了两个城市，price符合输入规则 |
| 后置条件 | 将两城市价格信息加入数据库或更新数据库已有信息，返回添加结果 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| cityDataSer.add(cityPO po) | 新增城市对象 | |
| cityDataSer.update(cityPO po) | 更新城市对象信息 | |
| DatabaseFactory.getCityDatabase | 得到city数据库的服务的引用 | |

accountLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Account.newCount | 语法 | public Boolean newCount(CountVO count); |
| 前置条件 | 用户新建一套账 |
| 后置条件 | 将新增的一套账加入数据库，返回加入是否成功 |
| Account.checkInitInfo | 语法 | public ResultMessage checkInitInfo(); |
| 前置条件 | 用户选择查看期初信息 |
| 后置条件 | 返回期初信息 |
| Account.addAccount | 语法 | public ResultMessage addAccount(AccountVO account); |
| 前置条件 | 用户输入了新账户信息 |
| 后置条件 | 将新增的账户信息加入数据库，返回加入结果 |
| Account.deleteAccount | 语法 | public ResultMessage deleteAccount(String id) |
| 前置条件 | 用户选择一个账户进行删除 |
| 后置条件 | 根据账户名在数据库中删除该账户信息，返回删除结果 |
| Account.reviseAccount | 语法 | public ResultMessage reviseAccount(AccountVO account) |
| 前置条件 | 修改了账户信息并确认 |
| 后置条件 | 更新数据库中该账户的信息，返回更新结果 |
| Account.accountList | 语法 | public ResultMessage accountList() |
| 前置条件 | 用户选择查看账户列表 |
| 后置条件 | 返回数据库中所有账户的列表 |
| Account.checkAccount | 语法 | public ResultMessage checkAccount() |
| 前置条件 | 用户选择一个账户进行查看 |
| 后置条件 | 返回该账户信息 |
| Account.searchAccount | 语法 | public ResultMessage searchAccount(String keywords) |
| 前置条件 | 用户检索账户并输入关键字 |
| 后置条件 | 根据该关键字查找符合条件的账户并返回 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| accountDataSer.findAccount(String keywords) | 在数据库中查找含有关键词的账户，返回数据 | |
| accountDataSer.addAccount(AccountPO  po) | 在数据库中新增一个账户 | |
| accountDataSer.deleteAccount(String id) | 在数据库中删除一个账户 | |
| accountDataSer.reviseAccount(AccountPO po) | 在数据库中修改一个账户的信息 | |
| accountDataSer.checkAccount(String id) | 在数据库中找到一个账户的信息，返回数据 | |
| DatabaseFactory.getAccountbase() | 得到account数据库的服务的引用 | |
| accountDataSer.addCount(countPO po) | 在数据库中新增一套账 | |
| accountDataSer.findInitInfo(String id) | 根据id在数据库中查找期初信息，返回数据 | |

stockLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Stock.setWarning | 语法 | public ResultMessage setWarning(int w, int part, String id) |
| 前置条件 | 用户输入了警戒线比例，选择了区 |
| 后置条件 | 修改数据库中的警戒线比例，并返回修改是否成功 |
| Stock.checkWarning | 语法 | public boolean checkWarning(String id) |
| 前置条件 | 用户新建了入库单 |
| 后置条件 | 对比数据库中库存数/库存容量和警戒线比例，返回是否报警 |
| Stock.checkStock | 语法 | public ResultMessage checkStock(Date start, Date end, String id) |
| 前置条件 | 用户输入了开始时间和结束时间 |
| 后置条件 | 从数据库中获取开始时间和结束时间之间的出入库情况，计算合计，统计数据，返回结果 |
| Stock.checkNow | 语法 | public ResultMessage checkNow(String id) |
| 前置条件 | 用户选择了库存盘点 |
| 后置条件 | 从数据库中获取当前库存情况，返回结果 |
| Stock.outputExcel | 语法 | public boolean outputExcel(String location,StockVO stock) |
| 前置条件 | 用户选择导出库存情况和导出位置 |
| 后置条件 | 将生成的Excel导出到相应位置，返回导出是否成功 |
| Stock.shelfList | 语法 | public ResultMessage shelfList(String id) |
| 前置条件 | 用户选择了库存分区 |
| 后置条件 | 读取数据库中的架信息，返回架信息 |
| Stock.addShelf | 语法 | public boolean addShelf(ShelfVO shelf, String id) |
| 前置条件 | 用户输入了新建的架的所有信息，确认新建 |
| 后置条件 | 在数据库中添加新建的架的信息，返回是否新建成功 |
| Stock.deleteShelf | 语法 | public boolean deleteShelf(String shelfId, String id) |
| 前置条件 | 用户选择了一个架，选择删除 |
| 后置条件 | 在数据库中去除相应架的信息，返回是否删除成功 |
| Stock.reviseShelf | 语法 | public boolean reviseShelf(ShelfVO shelf, String id) |
| 前置条件 | 用户修改了架信息，并确认 |
| 后置条件 | 在数据库中修改相应架的信息，返回是否修改成功 |
| Stock.checkShelf | 语法 | public ResultMessage checkShelf(String shelfId, String id) |
| 前置条件 | 用户选择了架，选择查看 |
| 后置条件 | 在数据库中找到相应架的信息，返回信息 |
| Stock.getWarning | 语法 | public ResultMessage getWarning(String id) |
| 前置条件 | 用户选择了库存报警 |
| 后置条件 | 在数据库中获得各区警戒线比例，返回值 |
| Stock.setPoint | 语法 | public boolean setPoint(String id, Date date) |
| 前置条件 | 用户确认盘点 |
| 后置条件 | 在数据库中增加一个盘点截止点，返回是否成功 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| DatabaseFactory.getStockData() | 得到Stock数据库的服务的引用 | |
| stockDataSer.getWarning(String id) | 从数据库中获得ID相应仓库的警戒线比例 | |
| stockDataSer.setWarning(String id, int w, int part) | 在数据库中设置ID相应仓库和相应区的警戒线比例 | |
| stockDataSer.getInOut(Date start, Date end, String id) | 在数据库中获得ID相应仓库开始时间和结束时间之间的出入库单 | |
| stockDataSer.getStock(String id) | 在数据库中获得ID相应仓库当前库存的情况 | |
| stockDataSer.getShelves(String id) | 在数据库中获得ID相应仓库当前架的情况 | |
| stockDataSer.addShelf(ShelfPO po, String id) | 在数据库中ID相应仓库添加一个架的信息 | |
| stockDataSer.deleteShelf(String shelfId, String id) | 在数据库中ID相应仓库删除一个架的信息 | |
| stockDataSer.reviseShelf(ShelfPO po, String id) | 在数据库中ID相应仓库修改一个架的信息 | |
| stockDataSer.getShelf(String shelfID, String id) | 在数据库中获得ID相应仓库一个架的信息 | |
| stockDataSer.setPoint(String id, Date date) | 在数据库中增加一个盘点截止点 | |

checkLogic模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Check.seeDiary | 语法 | public ResultMessage seeDiary(Date start, Date end) |
| 前置条件 | 输入的日期符合输入规则 |
| 后置条件 | 根据输入的开始和结束日期，返回相应的日志记录 |
| Check.seeStatementSheet | 语法 | public ResultMessage seeStatementSheet(Date start, Date end) |
| 前置条件 | 输入的日期符合输入规则 |
| 后置条件 | 根据输入的开始和结束日期，返回由入款单和收款单信息组成的经营情况表 |
| Check.seeCollectRecord | 语法 | public ResultMessage seeCollectRecord(Date date, String id) |
| 前置条件 | 输入了日期，选择了营业厅 |
| 后置条件 | 根据输入的日期，返回当日的收款记录 |
| Check.total | 语法 | public ResultMessage total(ArrayList<IncomeVO> income) |
| 前置条件 | 用户选择合计收款 |
| 后置条件 | 返回合计的收款总额 |
| Check.getHall | 语法 | public ResultMessage getHall() |
| 前置条件 | 用户选择查看经营情况表 |
| 后置条件 | 返回所有营业厅的名称 |
| Check.outputExcel | 语法 | public boolean outputExcel(ArrayList<IncomeVo> income, String location) |
| 前置条件 | 用户选择导出经营情况表 |
| 后置条件 | 导出经营情况表到相应地址 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| checkDataSer.getDiary(Date start, Date end) | 根据开始和结束日期从数据库中获得日志记录，返回数据 | |
| checkDataSer.getCollects(Date start, Date end) | 根据开始和结束日期从数据库中获得收款单记录，返回数据 | |
| checkDataSer.getCollectRec (Date date, String id) | 根据给的日期和营业厅编号从数据库中获得当日该营业厅收款记录，返回数据 | |
| checkDataSer.getHall() | 返回所有营业厅的编号和名称 | |
| DatabaseFactory.getDiaryDatabase | 得到diary数据库的服务的引用 | |
| DatabaseFactory.getRecordDatabase | 得到record数据库的服务的引用 | |

orderApprove模块接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| orderApprove.askExamine | 语法 | public ResultMessage askExamine() |
| 前置条件 | 用户请求审批单据 |
| 后置条件 | 返回待审批单据列表 |
| orderApprove.examine | 语法 | public boolean examine(boolean approve) |
| 前置条件 | 用户决定单据是否通过审批 |
| 后置条件 | 返回审批结果 |
| orderApprove.chooseOrder | 语法 | public ResultMessage chooseOrder(String id) |
| 前置条件 | 用户选择一个订单，查看详情 |
| 后置条件 | 系统返回订单详情 |
| 需要的服务（需接口） | | |
| 服务名 | 服务 | |
| orderApproveDataSer.update(  ArrayList<OrderPO> po) | 更新订单审批状态 | |
| orderApproveDataSer.getOrder(String id) | 从数据库中获取相应单据的信息，返回。 | |
| DatabaseFactory.getOrderDatabase | 得到orders数据库的服务的引用 | |