4.3.5 OrderNewUI模块

（1）模块概述

OrderNewUI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。

OrderNewUI模块的职责及接口参见软件系统结构描述文档5.2.2表CheckUI模块的接口规范。

（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LogicService. OrderNewLogic接口。OrderVO是各种单据作为单据记录的持久化对象被添加到设计模型中去的。 ExpressPanel，ReceivePanel，PaymentPanel，HallLoadingPanel，IncomePanel，RecipientPanel，SendPanel，ArrivePanel，CenterLoadingPanel，TransitPanel，StockInPanel，StockOutPanel用于显示对应的单据信息。FunctionFrame负责由功能主界面到具体界面的跳转。

OrderNewUI模块的设计如图所示



OrderNewUI模块各个类的职责如下表所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| FunctionFrame | 负责用户功能主界面到具体功能界面的跳转 |
| ExpressPanel | 负责显示快递单填写界面 |
| ReceivePanel | 负责显示收件单填写界面 |
| PaymentPanel | 负责显示付款单填写界面 |
| HallLoadingPanel | 负责显示营业厅装车单填写界面 |
| IncomePanel | 负责显示收款单填写界面 |
| RecipientPanel | 负责显示接收单填写界面 |
| SendPanel | 负责显示派件单填写界面 |
| ArrivePanel | 负责显示到达单填写界面 |
| CenterLoadingPanel | 负责显示中转中心装车填写界面 |
| TransitPanel | 负责显示中转单填写界面 |
| StockInPanel | 负责显示入库单填写界面 |
| StockOutPanel | 负责显示出库单填写界面 |

（3）模块内部接口规范

ExpressPanel的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| ExpressPanel.submit | 语法 | | public Boolean submit() |
| 前置条件 | | 用户选择生成快递单 |
| 后置条件 | | 将用户填写的VO数据传入逻辑层 |
| ExpressPanel.cancel | 语法 | | public Boolean cancel() |
| 前置条件 | | 用户选择取消创建快递单 |
| 后置条件 | | 系统回到上一次界面 |
| 需要的接口（需接口） | | | |
| 服务名 | | 服务 | |
| OrderNewSer.Create | | 负责提供新建单据的逻辑接口 | |

ReceivePanel的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| ReceivePanel.submit | 语法 | | public Boolean submit() |
| 前置条件 | | 用户选择生成接收单 |
| 后置条件 | | 将用户填写的VO数据传入逻辑层 |
| ReceivePanel.showDate | 语法 | | public ResultMessage showDate() |
| 前置条件 | | 用户打开界面 |
| 后置条件 | | 自动获取当前日期并显示在订单中 |
| ReceivePanel.cancel | 语法 | | public Boolean cancel() |
| 前置条件 | | 用户选择取消创建快递单 |
| 后置条件 | | 系统回到上一次界面 |
| 需要的接口（需接口） | | | |
| 服务名 | | 服务 | |
| OrderNewSer.Create | | 负责提供新建单据的逻辑接口 | |

PaymentPanel的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| PaymentPanel.submit | 语法 | | public Boolean submit() |
| 前置条件 | | 用户选择生成付款单 |
| 后置条件 | | 将用户填写的VO数据传入逻辑层 |
| PaymentPanel.showDate | 语法 | | public ResultMessage showDate() |
| 前置条件 | | 用户打开界面 |
| 后置条件 | | 自动获取当前日期并显示在订单中 |
| 后置条件 | | 自动获取显示订单到达地信息 |
| PaymentPanel.cancel | 语法 | | public Boolean cancel() |
| 前置条件 | | 用户选择取消创建快递单 |
| 后置条件 | | 系统回到上一次界面 |
| 需要的接口（需接口） | | | |
| 服务名 | | 服务 | |
| OrderNewSer.Create | | 负责提供新建单据的逻辑接口 | |

HallLoadingPanel的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| HallLoadingPanel.submit | 语法 | | public Boolean submit() |
| 前置条件 | | 用户选择生成营业厅接收单 |
| 后置条件 | | 将用户填写的VO数据传入逻辑层 |
| HallLoadingPanel.showDate | 语法 | | public ResultMessage showDate() |
| 前置条件 | | 用户打开界面 |
| 后置条件 | | 自动获取当前日期并显示在订单中 |
| HallLoadingPanel.showCities | 语法 | | public ResultMessage showCities() |
| 前置条件 | | 用户填写城市 |
| 后置条件 | | 自动获取显示城市列表 |
| HallLoadingPanel.cancel | 语法 | | public Boolean cancel() |
| 前置条件 | | 用户选择取消创建快递单 |
| 后置条件 | | 系统回到上一次界面 |
| 需要的接口（需接口） | | | |
| 服务名 | | 服务 | |
| OrderNewSer.Create | | 负责提供新建单据的逻辑接口 | |
| OrderNewSer.getCities | | 负责获取城市选项的逻辑接口 | |

IncomePanel的接口规范

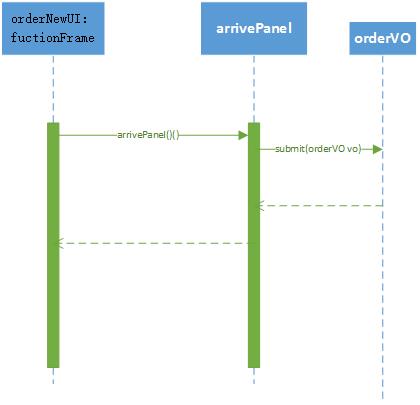
|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| IncomePanel.submit | 语法 | | public Boolean submit() |
| 前置条件 | | 用户选择生成收款单 |
| 后置条件 | | 将用户填写的VO数据传入逻辑层 |
| IncomePanel.showDate | 语法 | | public ResultMessage showDate() |
| 前置条件 | | 用户打开界面 |
| 后置条件 | | 自动获取当前日期并显示在订单中 |
| ReceivePanel.cancel | 语法 | | public Boolean cancel() |
| 前置条件 | | 用户选择取消创建快递单 |
| 后置条件 | | 系统回到上一次界面 |
| 需要的接口（需接口） | | | |
| 服务名 | | 服务 | |
| OrderNewSer.Create | | 负责提供新建单据的逻辑接口 | |
| OrderNewSer.getCourier | | 负责获取快递员选项的逻辑接口 | |

其余单据界面同上，不再赘述。

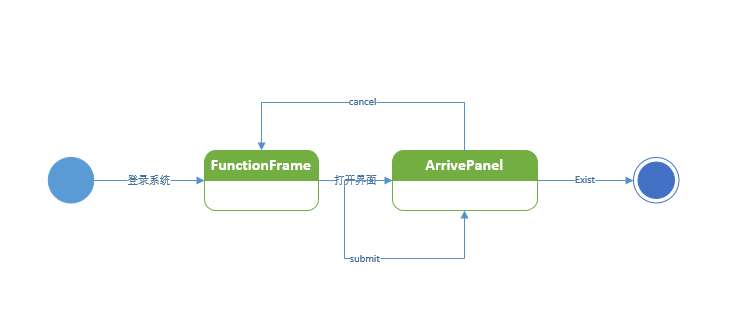
（4）界面层的动态模型

下图表明了系统中，当用户填写了接收单后，订单界面间的跳转。其余单据动态模型类似如图。

填写接收单的顺序图



填写接收单的状态图



4.3.5 stockUI模块

（1）模块概述

stockUI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。

stockUI模块的职责及接口参见软件系统结构描述文档5.2.2表stockUI模块的接口规范。

（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LogicService. stockLogic接口。StockVO是作为库存记录的持久化对象，ShelfVO是作为货架记录的持久化对象被添加到设计模型中去的。StockCheckNowPanel，StockCheckPanel，StockWarningPanel，StockDividePanel用于显示对应的单据信息。FunctionFrame负责由功能主界面到具体界面的跳转。

stockUI模块的设计如图所示



stockUI模块各个类的职责如下表所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| FunctionFrame | 负责用户功能主界面到具体功能界面的跳转 |

|  |  |
| --- | --- |
| StockCheckNowPanel | 库存查看界面，负责显示库存当前信息 |
| StockCheckPanel | 库存查看界面，负责显示库存信息 |
| StockWarningPanel | 库存警告界面，负责显示库存警告信息界面 |
| StockDividePanel | 库存分区界面，负责处理库存中架信息的管理操作 |

（3）模块内部接口规范

StockWarningPanel的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockWarningPanel.show | 语法 | public ResultMessage show() |
| 前置条件 | 用户打开警戒线修改界面 |
| 后置条件 | 显示原始警戒线值 |
| StockWarningPanel.change | 语法 | public Boolean change(String message) |
| 前置条件 | 用户选择提交库存警戒线 |
| 后置条件 | 改变库存警戒线的值 |
| StockWarningPanel.cancel | 语法 | public Boolean cancel() |
| 前置条件 | 用户选择取消库存警戒线改变 |
| 后置条件 | 界面显示原始库存警戒线的值 |
| StockWarningPanel.exception | 语法 | public void exception() |
| 前置条件 | 用户输入值越界 |
| 后置条件 | 报错并要求重新输入 |
| StockWarningPanel.exit | 语法 | public Boolean exit() |
| 前置条件 | 用户选择退出界面 |
| 后置条件 | 恢复至系统上一层界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| StockLogicSer.WarnSer | 负责提供库存报警服务的接口 |
| WarnSer.setWarning(int w, Part part, String id) | 负责设置警戒线 |
| WarnSer.checkWarning(String id) | 负责检测库存是否超出警戒线的接口 |
| WarnSer.getWarning(String id) | 负责获得警戒线 |

StockCheckPanel模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockCheckPanel.show | 语法 | public void show() |
| 前置条件 | 用户已经输入开始结束时刻 |
| 后置条件 | 显示时间段内库存信息 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| CheckSer.checkStock(Date start, Date end, String id) | 负责提供获得时间段中库存信息的接口 |
| StockLogicSer.CheckStockInSer | 负责提供查看入库服务 |
| StockLogicSer.CheckStockOutSer | 负责提供查看出库服务 |

StockCheckNowPanel模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockCheckNowPanel.checkNow | 语法 | public Boolean checkNow() |
| 前置条件 | 用户选择盘点，设置截止点 |
| 后置条件 | 进行库存盘点 |
| StockCheckNowPanel.sureCheck | 语法 | public Boolean sureCheck() |
| 前置条件 | 用户选择确认盘点 |
| 后置条件 | 显示导出excel地址的窗口 |
| StockCheckNowPanel.exit | 语法 | public Boolean exit() |
| 前置条件 | 用户选择退出 |
| 后置条件 | 回到系统上一层界面 |

|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |
| CheckSer.CheckNow | 负责提供库存盘点服务 |
| CheckNowSer.checkNow(String id) | 负责提供获得当前库存情况的接口 |
| CheckNowSer.setPoint(String id,Date date) | 负责提供设置截止点的接口 |
| CheckNowSer.outputExcel(String location, StockVO stock) | 负责提供导出库存情况的接口 |

StockDividePanel模块的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| StockDividePanel.search | 语法 | public Boolean search(String string) |
| 前置条件 | 用户输入架的编号，选择搜索 |
| 后置条件 | 显示相应架的分区信息 |
| StockDividePanel.add | 语法 | public Boolean add(String string) |
| 前置条件 | 用户输入信息并选择新增 |
| 后置条件 | 新增分区信息 |
| StockDividePanel.delete | 语法 | public Boolean delete(StockVO stock) |
| 前置条件 | 用户选择库存分区选项并选择删除 |
| 后置条件 | 删除分区信息 |
| StockDividePanel.change | 语法 | public Boolean change(String string) |
| 前置条件 | 用户输入信息并选择确认修改 |
| 后置条件 | 修改分区信息 |
| StockDividePanel.cancel | 语法 | public Boolean cancel() |
| 前置条件 | 用户选择取消修改 |
| 后置条件 | 显示原来分区信息 |
| StockDividePanel.exit | 语法 | public Boolean exit() |
| 前置条件 | 用户选择退出 |
| 后置条件 | 回到系统上一层界面 |

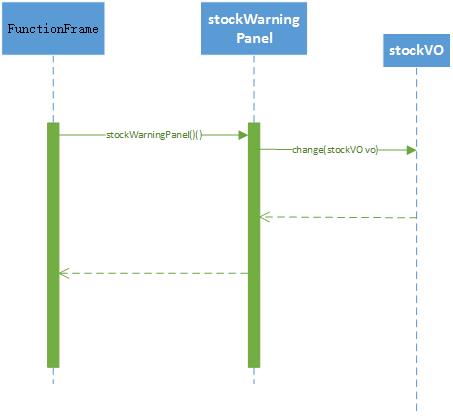
|  |  |
| --- | --- |
| 需要的接口（需接口） | |
| 服务名 | 服务 |

|  |  |
| --- | --- |
| StockLogicSer.DivideSer | 负责提供库存分区服务 |
| DivideSer.shelfList(String id) | 负责提供获得架列表的接口 |
| DivideSer.addShelf(ShelfVO shelf, String id) | 负责提供添加架的接口 |
| DivideSer.deleteShelf(String shelfId, String id) | 负责提供删除架的接口 |
| DivideSer.reviseShelf(ShelfVO shelf, String id) | 负责提供修改架的接口 |
| DivideSer.checkShelf(String shelfId, String id) | 负责提供查看架的接口 |

（4）界面层的动态模型

下图表明了系统中，当用户进行库存报警时界面间的跳转。

库存报警的顺序图



库存报警的状态图



库存查看的状态图



库存盘点，分区状态图类似上图。

4.3.7 TransitInfoUI模块

（1）模块概述

TransitInfoUI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。

TransitInfoUI模块的职责及接口参见软件系统结构描述文档5.2.2表TransitInfoUI模块的接口规范。

（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LogicService. transitInfoLogic接口。StockCheckNowPanel，StockCheckPanel，StockWarningPanel，StockDividePanel用于显示对应的单据信息。FunctionFrame负责由功能主界面到具体界面的跳转。

TransitInfoUI模块的设计如图所示



TransitInfoUI模块各个类的职责如下表所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| FunctionFrame | 负责用户功能主界面到具体功能界面的跳转 |

|  |  |
| --- | --- |
| TransitInfoInqPanel | 显示物流信息查询界面 |

（3）模块内部接口规范

TransitInfoUI模块的接口规范

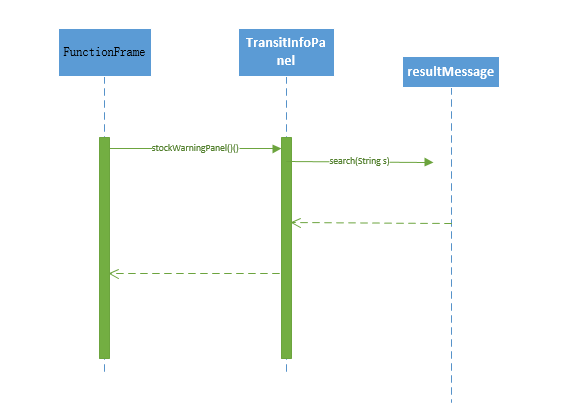
|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| TransitInfoInqPanel.search | 语法 | public ResultMessage search (String string) |
| 前置条件 | 用户输入订单编号 |
| 后置条件 | 显示物流信息 |
| 需要的接口（需接口） | | |

|  |  |
| --- | --- |
| TransitInfoLogicSer.enterBarcodeCounrier(String  Barcode) | 查询订单信息界面的逻辑接口 |
| TransitInfoLogicSer.enterBarcodeCustomer(String  Barcode) | 查询物流轨迹和货运状态界面的逻辑接口 |

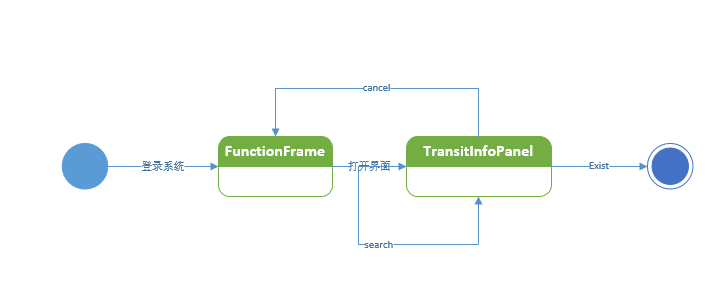
（4）界面层的动态模型

下图表明了系统中，当用户进行物流信息查询时界面间的跳转。

物流信息查询顺序图



物流信息查询状态图



4.3.8 UserManUI模块

（1）模块概述

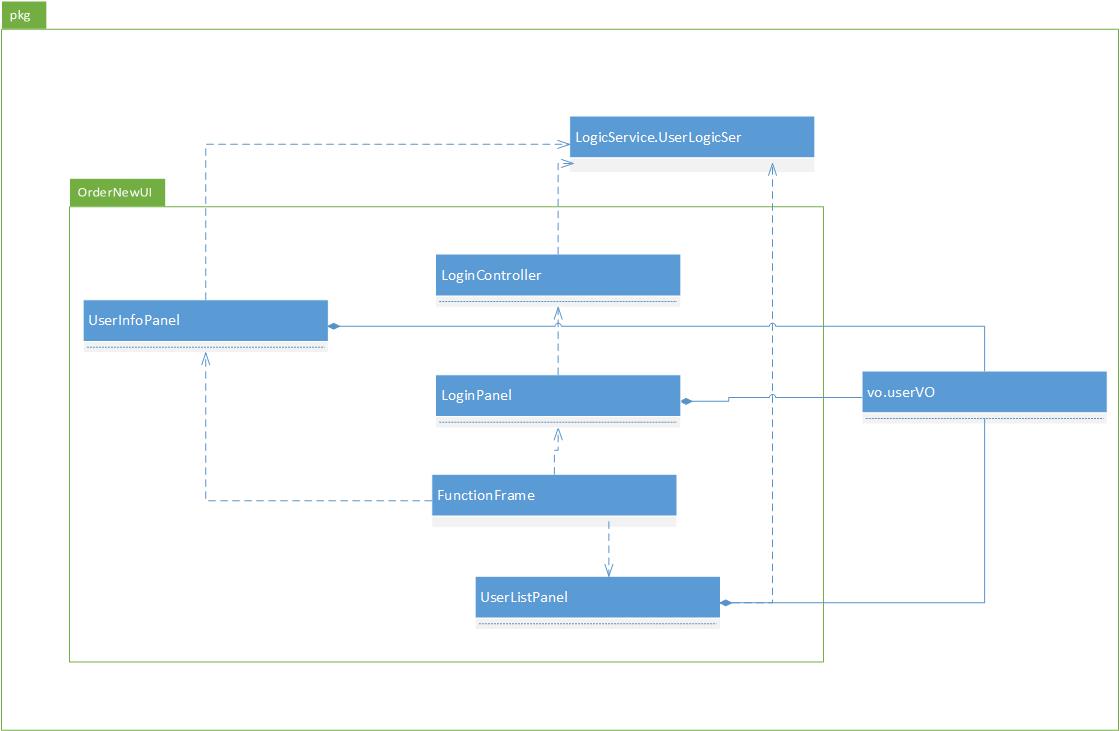
UserManUI模块承担的需求参见需求规格说明文档功能需求及相关非功能需求。

UserManUI模块的职责及接口参见软件系统结构描述文档5.2.2表UserManUI模块的接口规范。

（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LogicService. UserManLogic接口。为了隔离展示层职责和逻辑层职责，我们增加LoginController，将登陆的业务处理交给Login对象。AccountVO是作为账户记录的持久化对象，UserVO是作为用户记录的持久化对象被添加到设计模型中去的。UserInfoPanel，UserListPanel，LoginPanel用于显示对应的单据信息。FunctionFrame负责由功能主界面到具体界面的跳转。

UserManUI模块的设计如图所示



UserManUI模块各个类的职责如下表所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| FunctionFrame | 负责用户功能主界面到具体功能界面的跳转 |

|  |  |
| --- | --- |
| LoginPanel | 登录界面，负责处理登录操作 |
| UserInfoPanel | 用户信息界面，负责显示用户信息 |
| UserListPanel | 用户列表界面，负责显示所有的用户 |

（3）模块内部接口规范

LoginPanel模块的接口规范

UserUI模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| LoginPanel.login | 语法 | | public Boolean login(AccountVO vo) |
| 前置条件 | | 用户输入账号密码 |
| 后置条件 | | 显示登录结果 |
| LoginPanel.close | 语法 | | public Boolean close() |
| 前置条件 | | 用户选择关闭 |
| 后置条件 | | 关闭系统 |
| LoginPanel.search | 语法 | | Public TransitInfoInqPanel () |
| 前置条件 | | 用户选择物流查询 |
| 后置条件 | | 打开物流查询界面 |
| 需要的接口（需接口） | | | |
| 服务名 | | 服务 | |
| UserLogicSer.LogicSer | | 负责提供登录的逻辑接口 | |
| LogicSer.login(String id,String password) | | 负责处理登录的接口 | |

UserInfoPanel模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| UserInfoPanel.sure | 语法 | | public Boolean sure(AccountVO vo); |
| 前置条件 | | 用户确定信息 |
| 后置条件 | | 更新用户信息 |
| UserInfoPanel.close | 语法 | | public Boolean close (); |
| 前置条件 | | 用户请求关闭界面 |
| 后置条件 | | 关闭当前界面，回到上一层。 |
| 需要的接口（需接口） | | | |
| 服务名 | | 服务 | |
| UserLogicSer.UserSer | | 负责提供用户管理的逻辑接口 | |

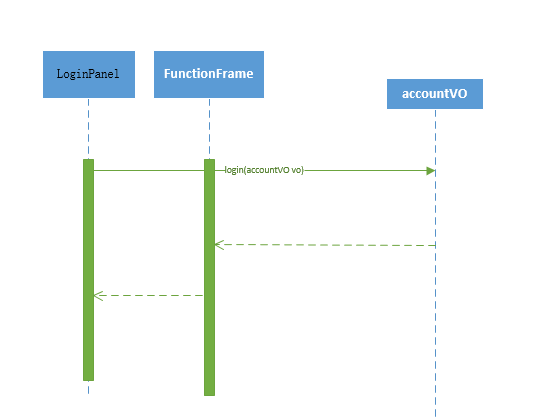
UserListPanel模块的接口规范

|  |  |  |  |
| --- | --- | --- | --- |
| 提供的服务（供接口） | | | |
| UserListPanel.search | 语法 | | public resultMessage search (String s); |
| 前置条件 | | 用户输入关键字并选择搜索 |
| 后置条件 | | 显示有相应关键字的用户 |
| UserListPanel.add | 语法 | | public Boolean add (); |
| 前置条件 | | 用户请求添加用户 |
| 后置条件 | | 打开添加用户窗口 |
| UserListPanel.delete | 语法 | | public Boolean delete(UserVO vo) |
| 前置条件 | | 用户选择删除选中用户 |
| 后置条件 | | 删除用户 |
| UserListPanel.change | 语法 | | public Boolean change(UserVO vo) |
| 前置条件 | | 用户选择修改选中用户 |
| 后置条件 | | 打开用户信息修改界面 |
| UserListPanel.check | 语法 | | public Boolean check(UserVO vo) |
| 前置条件 | | 用户选择查看选中用户 |
| 后置条件 | | 显示用户详细信息窗口 |
| 需要的接口（需接口） | | | |
| 服务名 | | 服务 | |
| UserSer.add(UserVO user) | | 负责新建用户的接口 | |
| UserSer.delete(String id) | | 负责删除用户的接口 | |
| UserSer.revise(UserVO user) | | 负责修改用户的接口 | |
| UserSer.check(String id) | | 负责查看用户的接口 | |
| UserSer.list() | | 负责获取用户列表的接口 | |

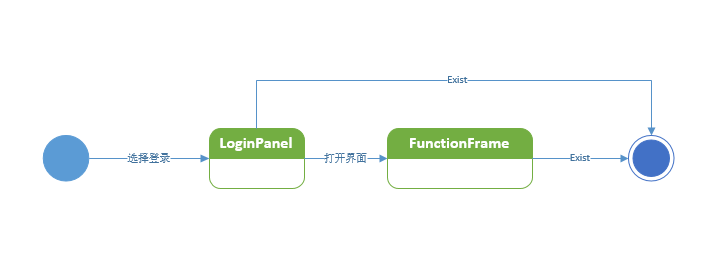
（4）界面层的动态模型

下图表明了系统中，当用户进行用户界面操作时界面间的跳转。

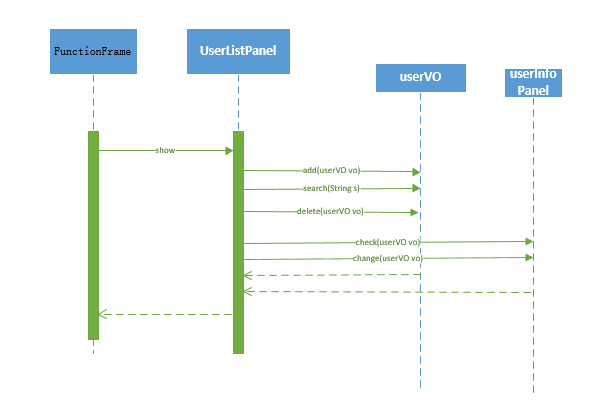
登录界面顺序图



登录界面状态图



用户信息管理界面顺序图



用户信息管理界面状态图

