《超市管理系统》设计说明书

VER 1.0

2016年5月

# 编写目的

此需求规格说明书对《超市管理系统》项目做了全面细致的用户需求分析，明确所要开发的软件应具有的功能、性能与界面，使开发人员能清楚地明白需求，并在此基础上进一步提出概要设计说明书和完成后续设计与开发工作。

本说明书定义超市管理系统的详细需求，明确了其功能内容、开发途径，是整个软件开发的依据，它对以后阶段的工作起指导作用。本文也是项目完成后系统验收的依据。

# 项目背景

超市管理系统具有商品管理、购物收银、超市管理等功能，本项目使用java作为开发语言，java控制台显示界面，sqlite数据库存储数据，可以有效地锻炼和加强学生运用java、sqlite数据库及基本sql编程开发的能力。

# 概要设计

本项目的目标是完成超市管理系统，主要功能有3个：商品管理、购物收银、超市管理。

1. 商品管理

实现对商品的增加、删除、修改、查询、罗列。

1. 购物收银

实现用户购物及结算收银。用户输入商品名称和购买数量，实时计算当前总额，结算时用户输入支付金额后，计算找零并更新库存商品数量，并将订单、详单插入数据库。本模块要求用户使用售货员账户登录。

1. 超市管理

包含3个子功能：人员管理、订单管理和详单管理。

1. 人员管理：对用户账户的增删改查。
2. 对订单的查询和显示。
3. 对详单的查询和显示。

本模块要求用户使用管理员账户登录。

1. 辅助功能

用户登录：对用户输入的用户名和密码进行校验，3次登陆机会。

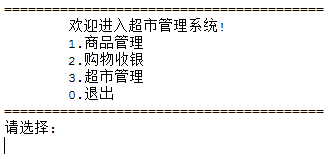
用户输入输出：获取用户输入的字符串、整数或浮点数。验证用户输入是否合法，如不合法则提示重新输入。以及对指定内容进行格式化输出。

格式化输出：封装system.out.println()方法，在输出内容前插入tab制表符。

# 详细设计

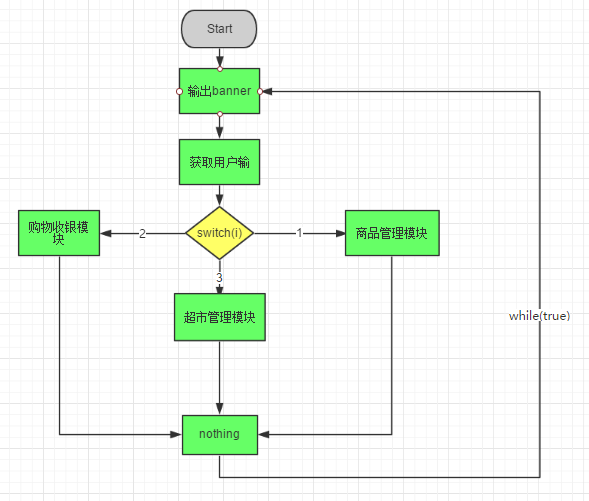
## 程序初始界面

程序起始界面设计为MainPage.java，主要功能为展示程序功能，并导航用户操作。界面效果如下图所示。



### mainPage()方法

展示程序的主界面，并根据用户的选择将其导向指定模块。流程图如下所示。



### outputBanner()方法

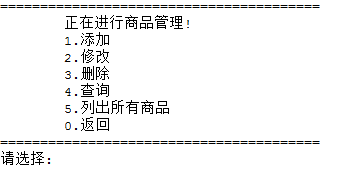
输出起始界面上的所有显示，包括横线和菜单项。为美观起见，各菜单项前添加1个制表符。具体的输出函数调用封装好的IOTools.outputWithTabs(String s)方法。

## 商品管理

本模块设计为GoodsMgmtPage.java，实现对商品的添加、修改、删除、查询、罗列5个功能，数据库操作调用GoodsDao.java实现。

### mainPage()方法

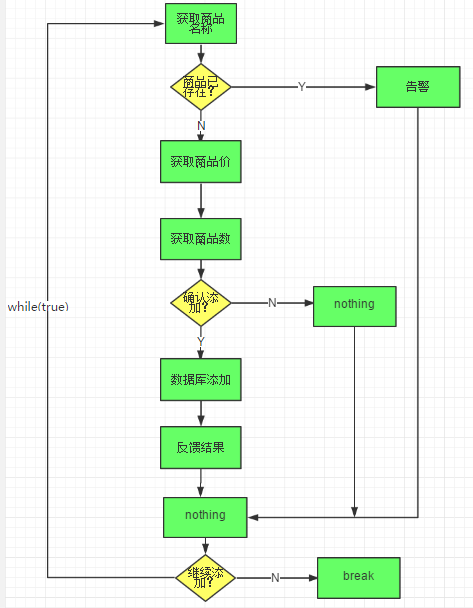
本方法功能为，展示菜单并导航用户操作。



实现流程图与MainPage.mainPage()方法基本相同。

### addGoods()方法

本方法实现商品的添加功能。本方法实现商品的添加功能。获取输入的商品名称后，首先在数据库中查询商品是否已经存在，如果已存在则给出提示并返回；如果不存在，则继续获取输入的商品数量和价格，在得到用户确认后进行数据库添加操作，并返回操作结果。最后，询问用户是否继续添加，选择是则重复以上流程，否则方法返回。业务流程如下图所示。



### modifyGoods()方法

本方法实现商品信息的修改功能，包括：名称、价格、数量。首先获取用户输入的商品名称并在数据库中查询，如果不存在则业务逻辑终止。接着，提示用户选择要修改的信息类别，最后获取用户输入的新值。根据用户的选择，调用数据库接口修改商品的指定信息。

### deleteGoods()方法

本方法实现商品的删除功能。用户输入商品名称后，首先查询商品是否存在，如不存在则业务逻辑终止。经确认后，在数据库中删除商品记录。

### queryGoods()方法

本方法实现商品的查询功能，可根据名称、价格和数量进行查询。其中，可对名称进行模糊查询。首先提示用户选择查询类别，然后输入关键字并返回查询结果。

### displayAllGoods()方法

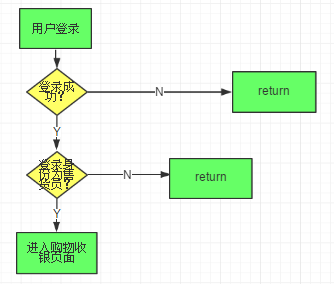
本方法实现商品的展示功能。可根据商品的数据库ID、名称、价格、数量进行增序或减序展示。首先提示用户选择查询类别和排序方式，然后调用对应的数据库接口进行查询，最后显示查询结果。

## 购物收银

本模块实现用户的购物结算功能。首先，必须以售货员的账户登录，否则无法进入收银界面。接着，不断要求用户输入商品名称和购买数量，并实时显示当前购物金额。当用户输入”zf”时，进入结算，显示购物总额。用户输入支付金额后，更新商品库存、订单、详单信息到数据库。如果更新成功则结算成功并找零，更新失败在结算失败并返回。

### mainPage()方法

本方法实现购物收银模块的界面入口和基本业务逻辑。首先，提示用户进行登录，并检查登录结果。如果登录成功，进一步检查登录身份，如果是以售货员登录，则进入购物收银界面。流程图如下所示。



### doCashier()方法

本方法实现购物收银功能。在一个while循环中，不断要求用户输入购买商品名称和数量，并查询数据库中的商品信息，判断输入是否合法。包括：该商品是否存在，库存数量是否足够。如果输入不合适则给出提示。检索商品时，可对名称进行模糊匹配，并输出匹配商品由用户选择。用户每次选择的商品均加入一个map，保存该商品的数据库id和购买数量。当用户输入”zf”时跳出循环进入结算。将对数据库中的商品数量更新、插入订单、插入详单操作作为一个事务进行执行。如果执行成功，则进行找零并提交更改；如果执行失败，则进行数据回滚并返回。

### getChoosedGoods()

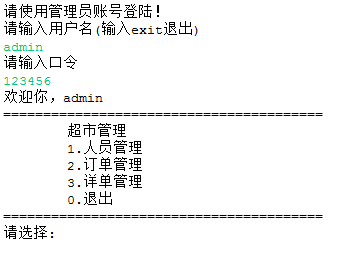
本方法为私有方法，用于获取用户指定的商品并返回。首先要求用户输入商品名称，如果在数据库中精确匹配不到，则进行模糊匹配，并返回所有匹配结果并由用户进行选择。最后返回用户所选择的商品，如果用户没有选择，则返回null。

## 超市管理

本模块实现超市管理功能，包括：人员管理、订单管理和详单管理。设计为MarketMgmtPage.java。

### mainPage()方法

本方法实现本模块的初始界面，显示菜单并导航用户操作。首先，提示用户使用管理员账户登录，如果登陆失败或账户权限不对则无法进入操作界面。在操作界面中，根据用户选择将其导入对应的功能模块。初始界面及菜单界面如下所示。



### userManage()方法

调用UserMgmtPage.mainPage()，进入人员管理功能。

### goodsOrderManage()方法

功能为显示所有订单，通过调用displayAllOrder()方法实现。

### displayAllOrder()方法

罗列所有订单。可以根据订单号、售货员ID、订单价格、日期按照升序或降序排列。根据用户选择，调用对应的数据库接口进行查询，最后显示查询结果。

### goodsOrderDetailManage()方法

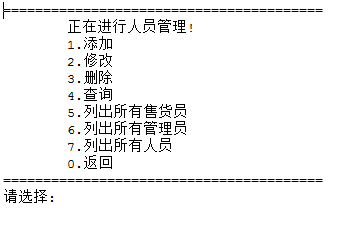
功能为根据详单号进行查询，通过调用GoodsOrderDetailDao().queryGoodsOrderDetail(i)方法实现。

## 人员管理

本模块实现人员（账户）的管理，包括添加、修改、删除、查询和罗列。人员分2类，售货员和管理员。本类设计为UserMgmtPage.java。

### mainPage()方法

本方法实现人员管理模块的主逻辑。输出菜单界面并导航用户操作。菜单界面如下所示。



### addUser()方法

添加用户。首先获取用户输入的用户名。如果该用户名已存在则提示并返回。接着，要求用户输入2次密码，如果密码不符合要求或不一致则提示重新输入。接着要求用户选择用户类型，最后经确认后添加数据库并返回结果。

### modifyUser()方法

修改用户信息。包括：用户名、密码。首先获取修改信息的类型，接着获取新值，最后修改数据库并返回结果。

### deleteUser()方法

删除用户。首先获取用户指定的用户，接着在数据库中删除记录并返回结果。

### queryUser()方法

查询用户。根据输入的用户名进行精确查找并返回匹配的用户。

### displayU**ser(UserType userType**)方法

罗列userType指定类型的所有用户。userType包括：售货员、管理员和所有人员。可根据数据库ID和姓名的升序和降序进行显示。首先提示用户选择排序内容和方式，接着调用对应的数据库接口进行查询，最后显示查询结果。

## 辅助功能

### 用户登录

本模块功能为用户登录界面的展示和登录验证。本类设计为LoginPage.java。

mainPage(**int** retry):实现本模块的界面展示和主逻辑。获取用户输入并调用验证方法。Retry指定了最大失败次数，失败达到该次数时，本方法返回false。

validateUser(String name, String password):本方法为私有方法，对给定的用户名和密码进行验证并返回结果。

### 用户输入输出

获取用户输入的字符串、整数或浮点数，当输入不合法时，给出提示并要求重新输入。本类设计为IOTools.java。

getInputString():获取用户输入的字符串。

getInputInt():获取用户输入的整数，如果输入不合法，则给出提示并要求重新输入。

getInputFloat():获取用户输入的浮点数，如果输入不合法，则给出提示并要求重新输入。

isChooseYes():获取用户输入的Y或N，如果输入不合法，则给出提示并要求重新输入。

outputWithTabs(String content)：对指定内容，前面加上制表符后输出。

outputBannerLine():输出banner中的横线。

本类中的输入验证使用正则表达式实现。

# 数据设计

## 数据库设计

数据库使用sqlite，它的特点是体积小，速度快。根据项目功能，设计使用4张表，如下所示。

|  |  |  |
| --- | --- | --- |
| 序号 | 表名称 | 用途 |
| 1 | GOODS | 商品表 |
| 2 | GOODS\_ORDER | 订单表 |
| 3 | GOODS\_ORDER\_DETAIL | 详单表 |
| 4 | USER | 人员表 |

商品表详细设计：

每一条商品信息包含4个信息：商品编号、名称、价格和数量，表结构如下所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 主键 | 说明 |
| 1 | GID | INTEGER PRIMARY KEY | Y | 商品编号,主键，自动生成 |
| 2 | GNAME | TEXT |  | 商品名称 |
| 3 | GPRICE | NUMERIC |  | 商品价格 |
| 4 | GNUM | NUMERIC |  | 商品数量 |

订单表详细设计：

订单主要包含4个元素：编号、售货员ID、金额和日期。表结构如下所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 主键 | 说明 |
| 1 | GOID | INTEGER PRIMARY KEY | Y | 订单编号,主键，自动生成 |
| 2 | SID | NUMERIC |  | 售货员ID |
| 3 | PRICE | NUMERIC |  | 订单金额 |
| 4 | DATE | TEXT |  | 订单日期 |

详单表详细设计：

详单表实际上是以订单编号为索引而存储的一张商品数据表，存储订单中的商品ID、商品出售价格和出售数量。表结构如下所示。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 主键 | 说明 |
| 1 | GODID | INTEGER PRIMARY KEY | Y | 详单编号,主键，自动生成 |
| 2 | GOID | NUMERIC |  | 订单编号 |
| 3 | GID | NUMERIC |  | 商品编号 |
| 4 | PRICE | NUMERIC |  | 商品价格 |
| 5 | NUMBER | NUMERIC |  | 销售数量 |

人员表详细设计：

每个人员具有4个属性：ID、名称、口令和类型。其中，类型有2种取值：售货员和管理员，由UserType枚举定义。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 列名 | 类型 | 主键 | 说明 |
| 1 | UID | INTEGER PRIMARY KEY | Y | 人员编号,主键，自动生成 |
| 2 | UNAME | TEXT |  | 人员名称 |
| 3 | UPASSWORD | TEXT |  | 人员口令 |
| 4 | UTYPE | TEXT |  | 人员类型 |

## 数据接口设计

### 数据库操作接口

本类封装了对sqlite数据库的直接操作，能够简化业务代码的编写。封装的方法包括：数据库的连接、断开、执行查询、执行操作、事务执行、事务提交、事务回滚。本类设计为SqlHelper.java类。

1. connect()方法

根据sqlite驱动及数据库路径连接数据库，并返回该连接。

1. dispose()方法

释放当前数据库连接和数据库语句资源。

1. executeUpdate(PreparedStatement preparedStatement)方法

使用preparedStatement所携带的sql及参数执行数据库更新操作。

1. executeQuery(PreparedStatement preparedStatement)方法

使用preparedStatement所携带的sql及参数执行数据库查询操作。

1. executeBatch(List<String> sqlList,

PreparedStatement preparedStatement)方法

使用preparedStatement和sqlList执行事务操作。

1. commit()方法

使用已保存的preparedStatement执行事务提交。

1. rollback()方法

使用已保存的preparedStatement执行事务回滚。

1. queryMaxValue(String colName, String tableName)方法

查询指定表中指定列的最大值。用于确定执行插入操作后的自增ID。

### 商品数据操作接口

本类封装了对商品数据的增删改查操作，能够简化业务代码的编写，内部调用数据库操作接口。设计为GoodsDao.java。

addGoods(Goods goods)方法

本方法插入一条商品信息到数据库。

modifyGoods(GoodsProperty gp, Goods goods)方法

本方法在数据库中更新商品信息，其中goods给出了商品的ID和最新数据，GoodsProperty 指定了待更新的字段。

deleteGoods(int id)方法

本方法在数据库中删除指定商品信息。

queryAllGoods(GoodsProperty gp, boolean isAscending)方法

本方法在数据库中查询所有商品信息，isAscending指定了排序，gp指定了排序字段。

fuzzyQueryByName(String name)方法

本方法根据名称在数据库中进行模糊查询，在sql语句中使用了like通配符。

queryGoodsByProperty(GoodsProperty gp, Object value)方法

本方法在数据库中查询所有匹配的商品，其中gp指定了匹配字段，value指定了字段值。

convertResultSet2List(ResultSet rs)方法

本方法为私有方法，用于将查询返回的rs中的数据提取为1个商品列表。

### 订单数据操作接口

本类封装了对订单数据的增删改查操作，能够简化业务代码的编写，内部调用数据库操作接口。设计为GoodsOrderDao.java。

addGoodsOrder(GoodsOrder go)方法

本方法插入一条订单信息到数据库。

deleteGoodsOrder(int id)方法

本方法在数据库中删除指定订单信息。

queryGoodsOrder(GoodsOrderProperty gop, boolean isAscending)方法

本方法在数据库中查询所有订单信息，isAscending指定了排序，gop指定了排序字段。

queryGoodsOrder(int id)

本方法在数据库中查询指定的订单信息。

### 详单数据操作接口

本类封装了对详单数据的操作，能够简化业务代码的编写，内部调用数据库操作接口。设计为GoodsOrderDaoDetail.java。

queryGoodsOrderDetail(int goid)方法

本方法根据订单号在数据库中查询详单信息。

### 人员数据操作接口

本类封装了对人员数据的增删改查操作，能够简化业务代码的编写，内部调用数据库操作接口。设计为UserDao.java。

addUser(User user)方法

本方法插入一条人员信息到数据库。

modifyUser(UserProperty gp, User user)方法

本方法在数据库中更新人员信息，其中user给出了人员的ID和最新数据，UserProperty 指定了待更新的字段。

deleteUser(int id)方法

本方法在数据库中删除指定人员信息。

queryAllUser(UserProperty gp, boolean isAscending)方法

本方法在数据库中查询所有人员信息，isAscending指定了排序，gp指定了排序字段。

queryUserByName(String name, boolean isFuzzy)方法

本方法根据名称在数据库中进行模糊或精确查询。isFuzzy为true时，使用like通配符执行模糊查询。

convertResultSet2List(ResultSet rs)方法

本方法为私有方法，用于将查询返回的rs中的数据提取为1个人员列表。

# 改进目标

为了使本项目更加实用，本项目可以在以下方面进行改进：

1. 数据库更换为Mysql、MSSql或Oracle等网络数据库。
2. 增强安全性，对密码进行加密处理。
3. 增加图形化操作界面。