目录

[功能说明 3](#_Toc11360961)

[技术选型 3](#_Toc11360962)

[一、实验课题一：人员档案的管理、物料档案管理 4](#_Toc11360963)

[1、人员档案管理 4](#_Toc11360964)

[1.0、主要需求 4](#_Toc11360965)

[1.1、数据库表设计 4](#_Toc11360966)

[1.2、主要流程图 5](#_Toc11360967)

[1.3、人员档案添加 5](#_Toc11360968)

[1.4、人员档案修改 8](#_Toc11360969)

[1.5、人员档案删除 10](#_Toc11360970)

[1.6、人员档案查询 12](#_Toc11360971)

[1.7 其他功能 14](#_Toc11360972)

[2、物料档案管理 16](#_Toc11360973)

[2.0、主要需求 16](#_Toc11360974)

[2.1、数据库表设计 17](#_Toc11360975)

[2.2、物料档案查询 17](#_Toc11360976)

[2.3、物料档案添加 19](#_Toc11360977)

[2.4、物料档案删除 22](#_Toc11360978)

[2.5、物料档案修改 25](#_Toc11360979)

[二、实验课题二，课题三：单/多物料进出仓信息管理 27](#_Toc11360980)

[1、进出仓管理 27](#_Toc11360981)

[1.0 主要需求 27](#_Toc11360982)

[1.1数据库表设计 28](#_Toc11360983)

[1.2物料进仓 29](#_Toc11360984)

[1.3 物料出仓 32](#_Toc11360985)

[1.4 物料订单搜索 35](#_Toc11360986)

[三、实验课题四：用户登录与权限设计 37](#_Toc11360987)

[1.0 主要需求 37](#_Toc11360988)

[1.1 数据库表设计 38](#_Toc11360989)

[1.2 用户登录 38](#_Toc11360990)

[1.3 权限设计 40](#_Toc11360991)

[四、实验课题五：统计程序设计与报表打印 41](#_Toc11360992)

[1.0 主要需求 41](#_Toc11360993)

[1.1 数据库表设计 42](#_Toc11360994)

[1.2 展示某个物料在某个时间段内进出仓折线图 42](#_Toc11360995)

[1.3 打印某个时间段所有进出仓订单信息 44](#_Toc11360996)

[1.4 打印某个物料某个时间段内进出仓订单信息 46](#_Toc11360997)

[五、实验课题六：应用更新 48](#_Toc11360998)

[1.0 主要需求 48](#_Toc11360999)

[1.1 实现一个文件上传服务器用于保存更新文件以及更新信息 48](#_Toc11361000)

[1.2 客户端与更新服务器端交互获取当前最新版本信息 50](#_Toc11361001)

[1.3 实现一个更新程序，用于下载最新客户端 51](#_Toc11361002)

[六、其他功能：日志管理 51](#_Toc11361003)

[1.0 主要需求 51](#_Toc11361004)

[1.1 数据库表设计 52](#_Toc11361005)

[1.2 利用切面编程添加日志 52](#_Toc11361006)

[1.3 日志查询 53](#_Toc11361007)

[七、总结 54](#_Toc11361008)

[八、团队工作质量自评表 54](#_Toc11361009)

[九、团队工作量自评表 57](#_Toc11361010)

[十、实验的源程序代码 62](#_Toc11361011)

[1、主要项目功能介绍 62](#_Toc11361012)

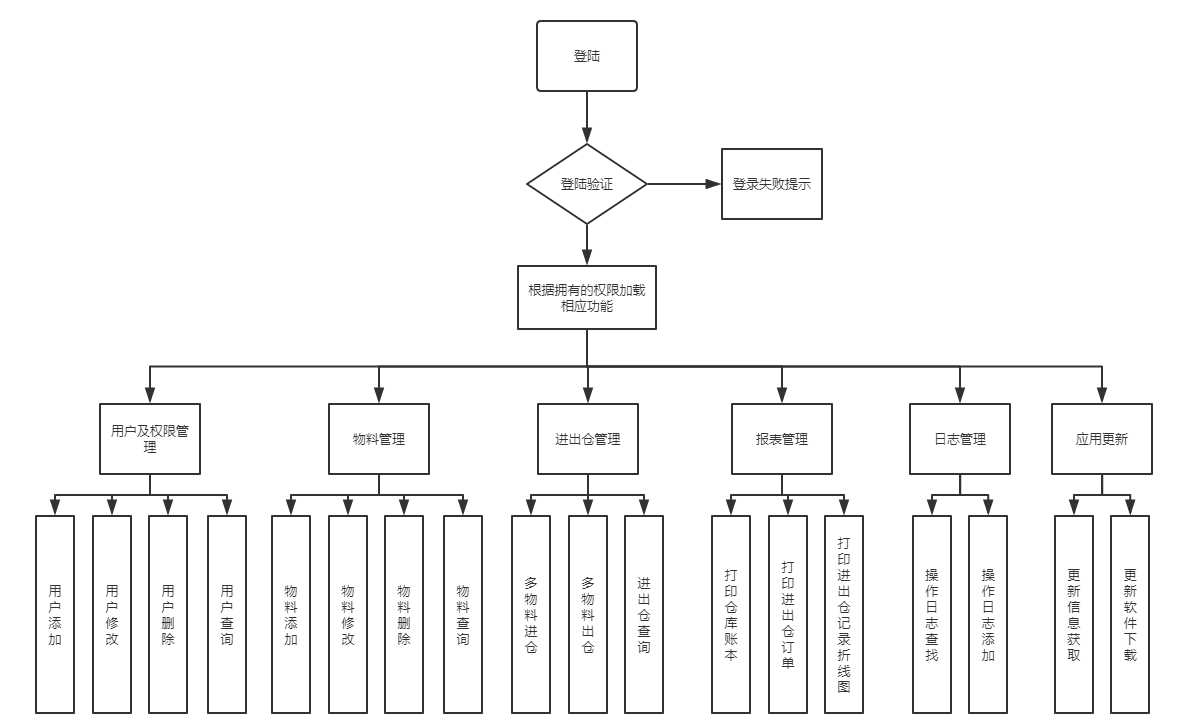
[2、工程文件结构： 63](#_Toc11361013)

[3、具体代码详见附带压缩包 63](#_Toc11361014)

# 功能说明

本应用实现了对人员的增加、删除、修改、查找以及权限的管理，以及对物料的增加、删除、修改、查找、进出仓以及打印报表的管理，为了记录人员操作，增加了日志系统，以及为了后续应用的扩展添加了应用更新功能。

总流程图为：



# 技术选型

1、开发语言： Java

2、数据库： MySQL

3、开发工具： Eclipse、SQLyog（数据库可视化工具）

4、主要技术栈： 1、Java Swing

2、SpringBoot

3、Mybatis

4、Spring

5、Maven

# 一、[实验课题一：人员档案的管理、物料档案管理](#_Toc23170)

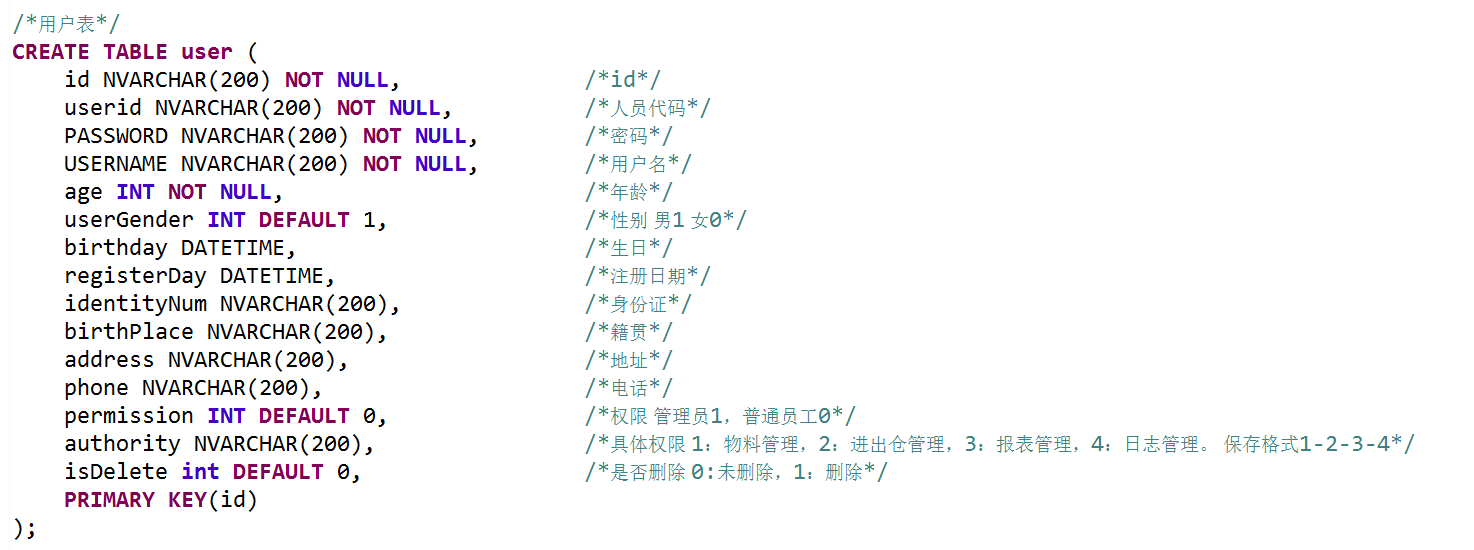
## 1、[人员档案管理](#_Toc29157)

### 1.0、主要需求

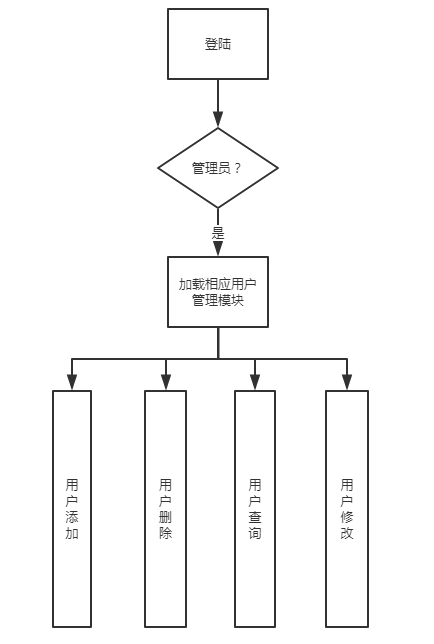
设计实现对企业人员基本情况的档案的输入、删除与修改，企业人员基本情况主要包括：人员代码、姓名、性别、出生日期、身份证号、籍贯、家庭住址、联系电话、其它情况等。

1. 程序须包括主界面程序，主界面可以通过菜单调用到其它所有子程序。
2. 人员档案的增加、修改、删除的程序，其中人员代码不能为空，性别必须用“男”、“女”选择方式，出生日期必须用日期控件来控制。修改、删除程序在执行修改删除操作前，都必须进行记录定位操作，所以必须有相应的查询程序，可以借用查询程序来完成记录定位。人员档案的增加、修改、删除程序也可以用三个程序名，分别编程。
3. 人员档案查询程序，可以在查询框中输入人员姓名中的任何一个单字，查询出含该字的所有人员。

### 1.1、数据库表设计

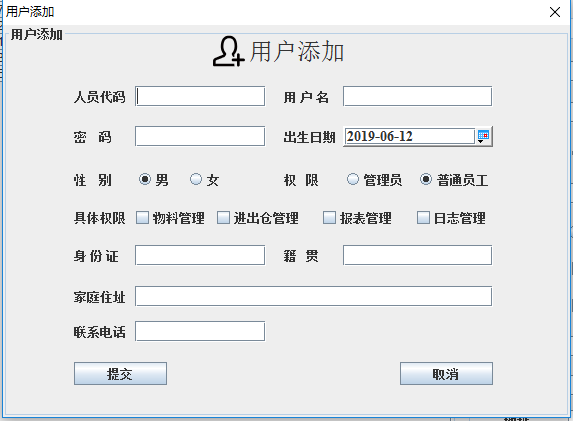


### 1.2、主要流程图



### 1.3、[人员档案添加](#_Toc14270)

#### 1.3.1 具体界面设计



#### 1.3.2 实现思路

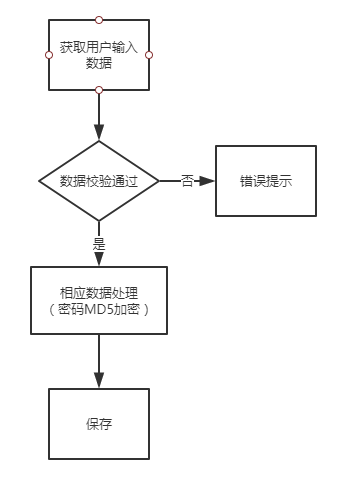
1、获取用户输入数据，对密码进行MD5加密，同时对于权限功能进行格式化处理

2、调用添加用户方法增加用户记录

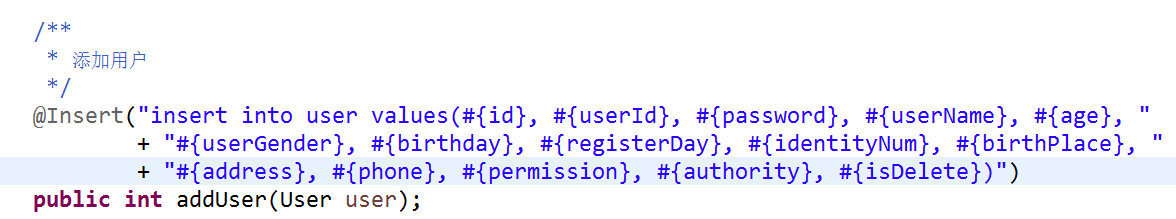
对于输入数据的具体要求：

1. 各个输入框不能为空
2. 出生日期应小于当前日期
3. 密码长度应在6-12位之间
4. 人员代码必须唯一

#### 1.3.3 具体流程图

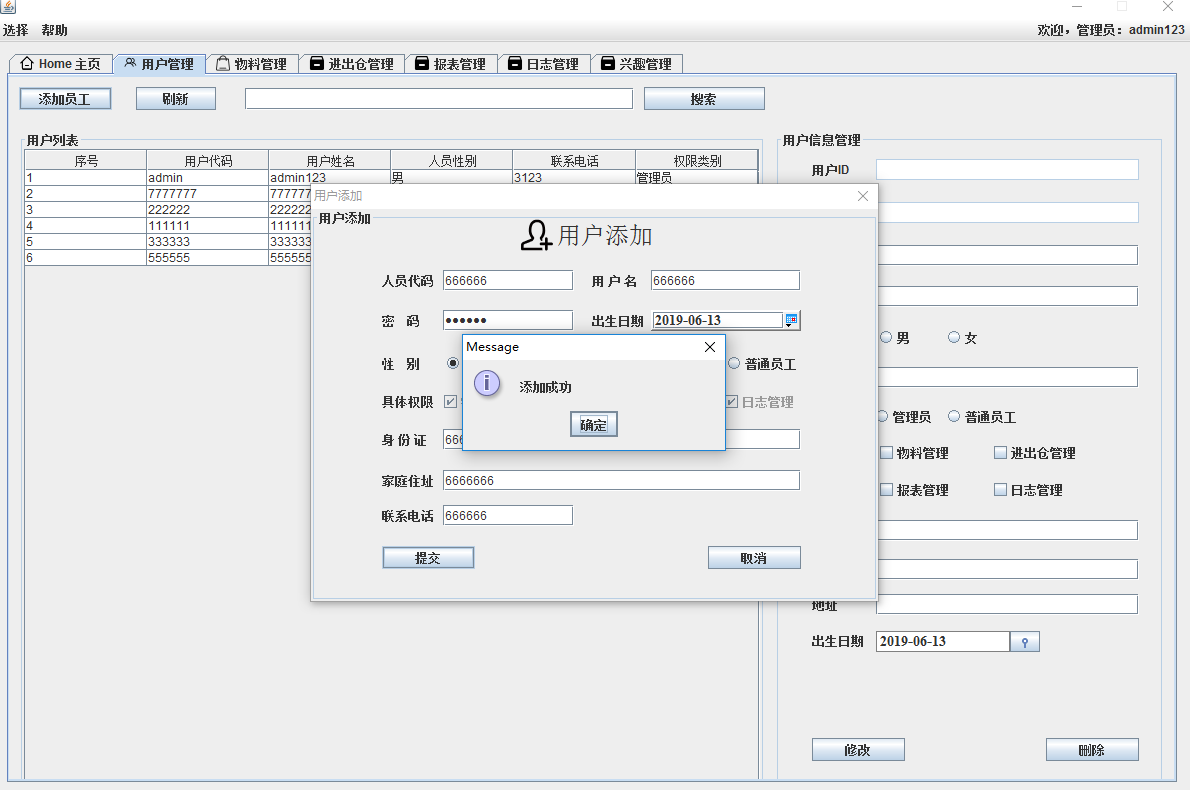


#### 1.3.4 主要数据库操作

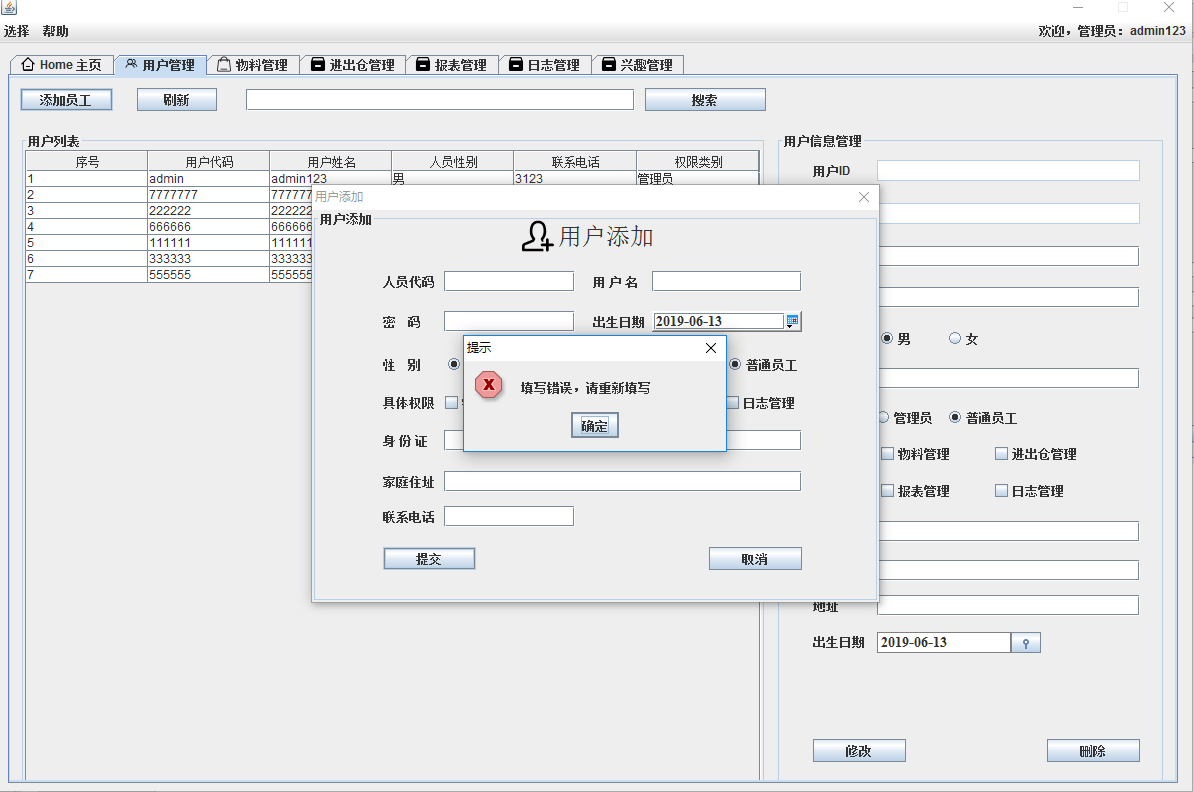


#### 1.3.5 主要测试

用户添加

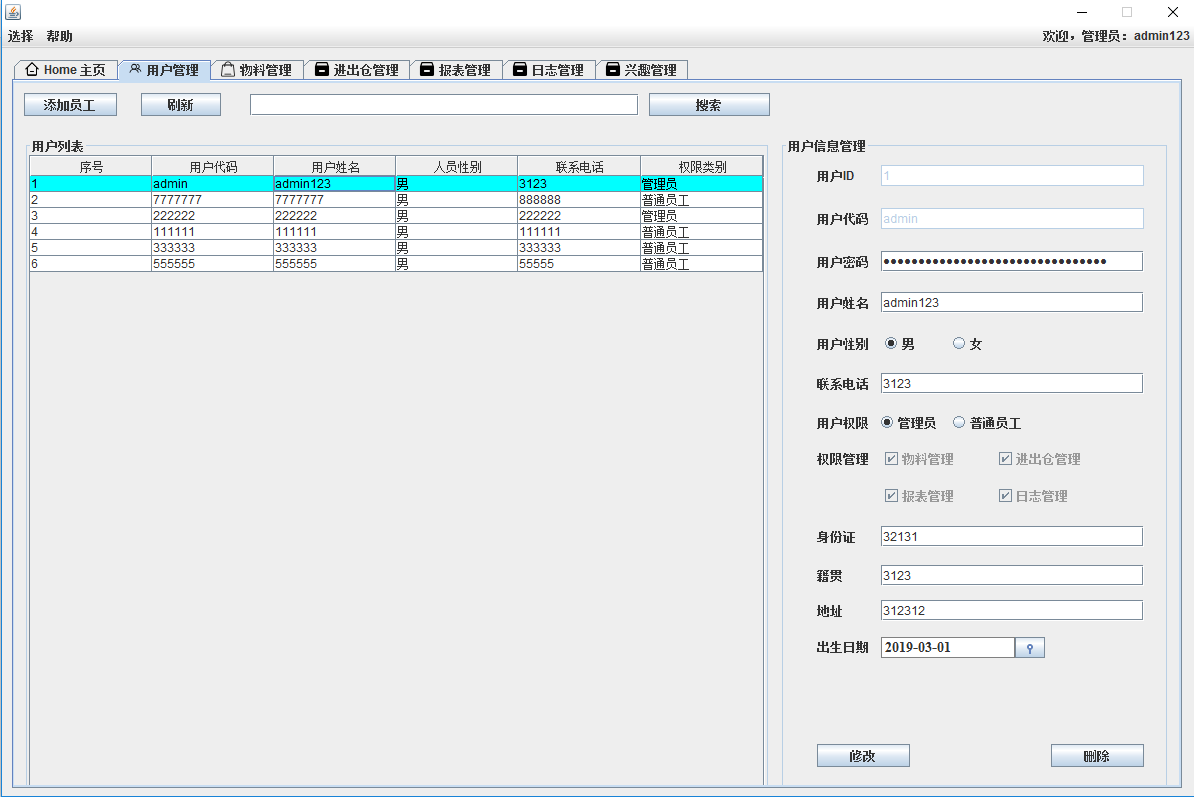


数据填写错误



### [1.4、人员档案修改](#_Toc20036)

#### 1.4.1 具体界面



#### 1.4.2 实现思路

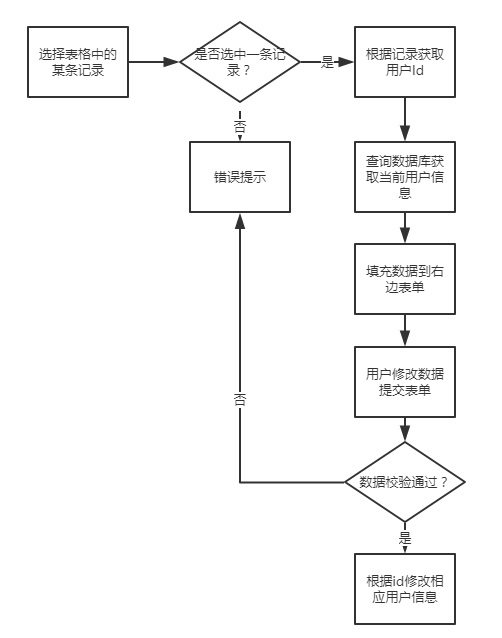
1、登录用户会将信息保存到Session缓存中，通过Session缓存获取到当前登陆用户信息，初始化用户修改界面，将用户数据填写到相应的文本框内。

2、用户修改信息，点击修改提交表单信息，判断密码是否是加密过的，如果密码是加密后（32位）则密码不变，若密码修改，则使用MD5进行加密。

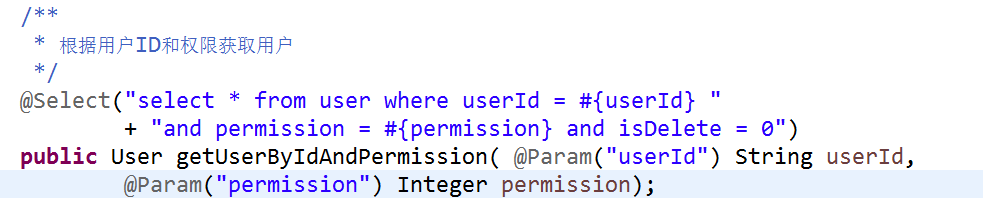
3、后台根据id和用户代码（userId）作为依据修改相应用户记录。

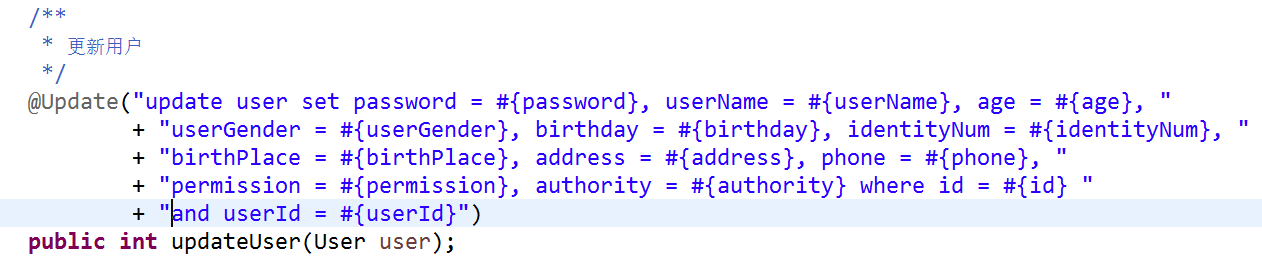
4、返回相应执行结果，展示提示界面。

#### 1.4.3 具体流程

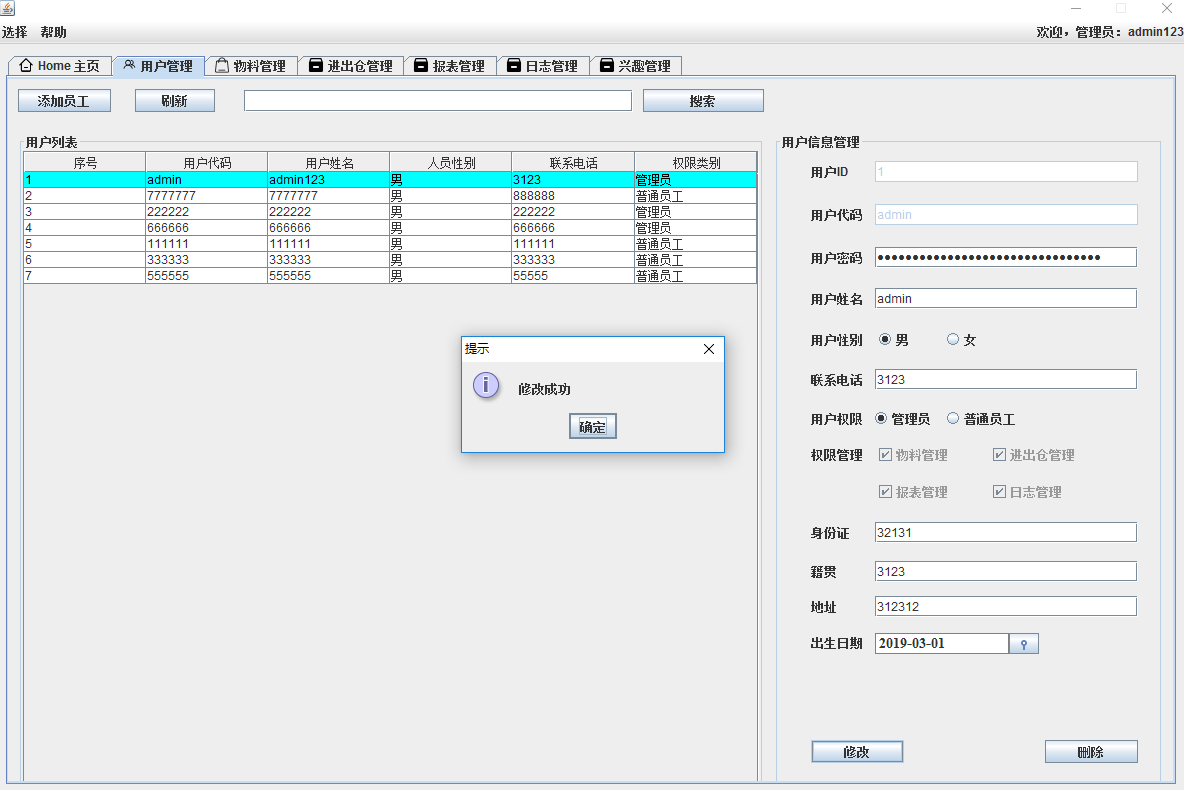


#### 1.4.4 主要数据库操作



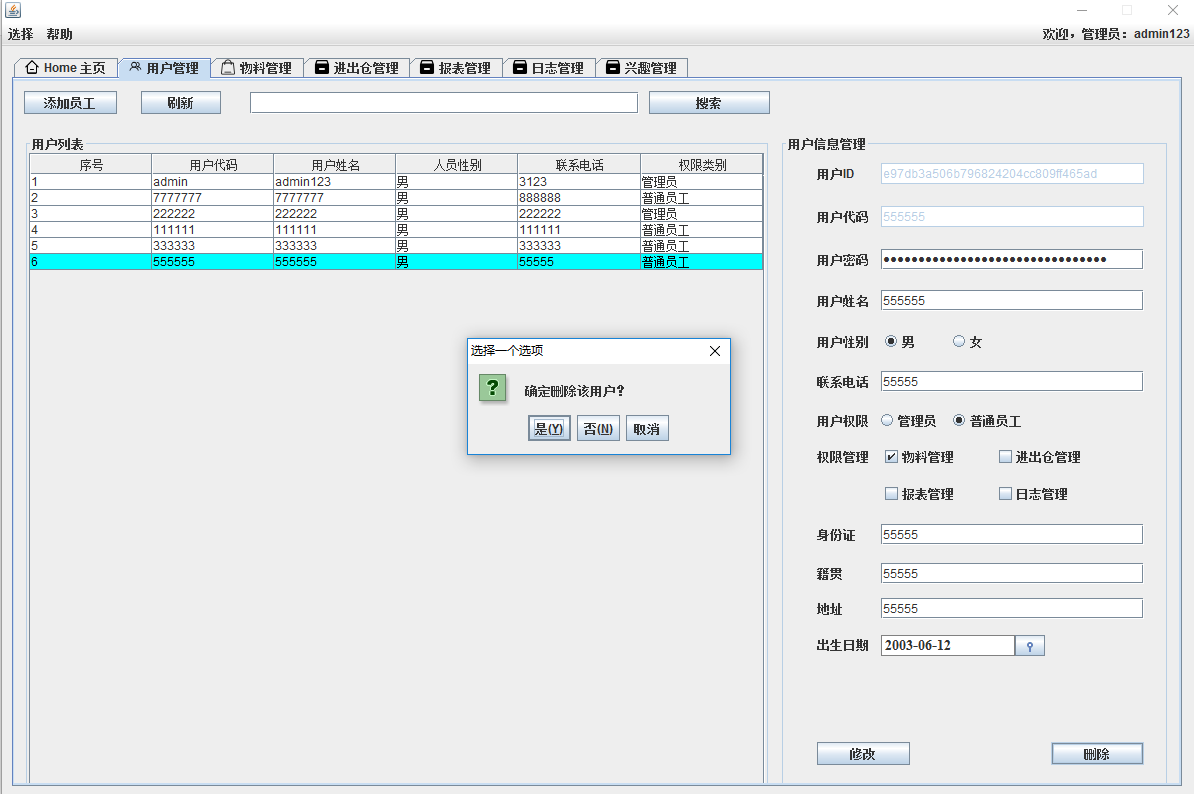


#### 1.4.5 主要测试



### [1.5、人员档案删除](#_Toc26011)

#### 1.5.1 具体界面



#### 1.5.2 实现思路

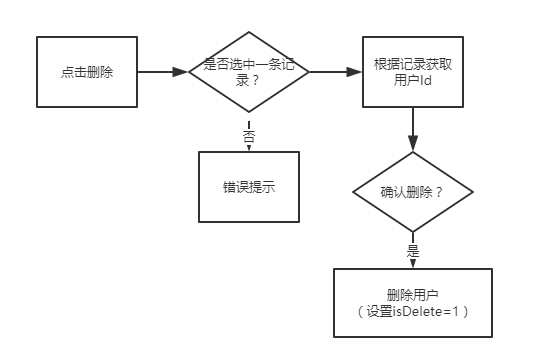
1、当选择某一行时会自动提取当前行的userId（用户代码）数据。

2、根据提取到的userId，在数据库中查询到相应用户的所有信息，并将查找到的信息填充到上图用户信息管理表单中。

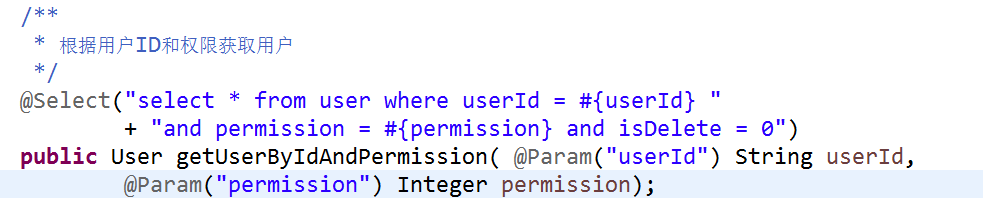
3、当用户点击删除时，会根据id和userId删除当前记录

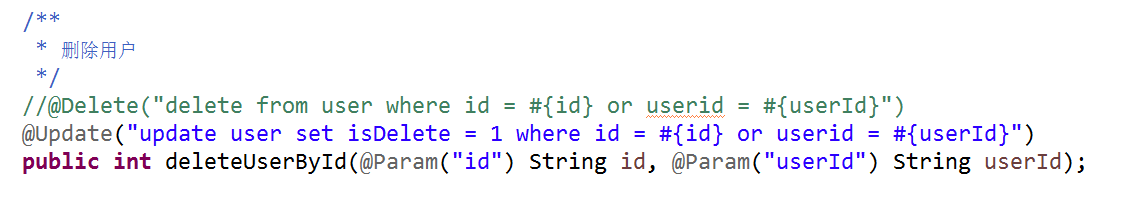
4、根据返回结果，显示提示信息

#### 1.5.3 具体流程

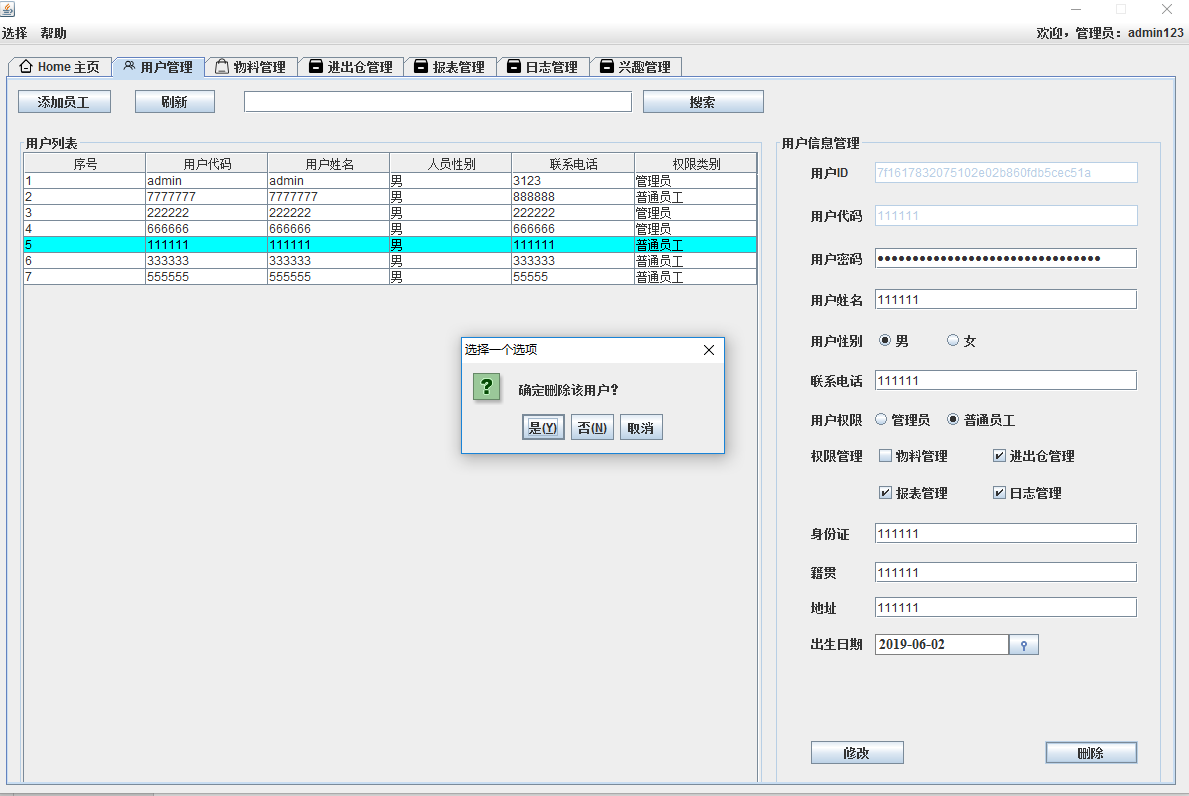


#### 1.5.4 主要数据库操作



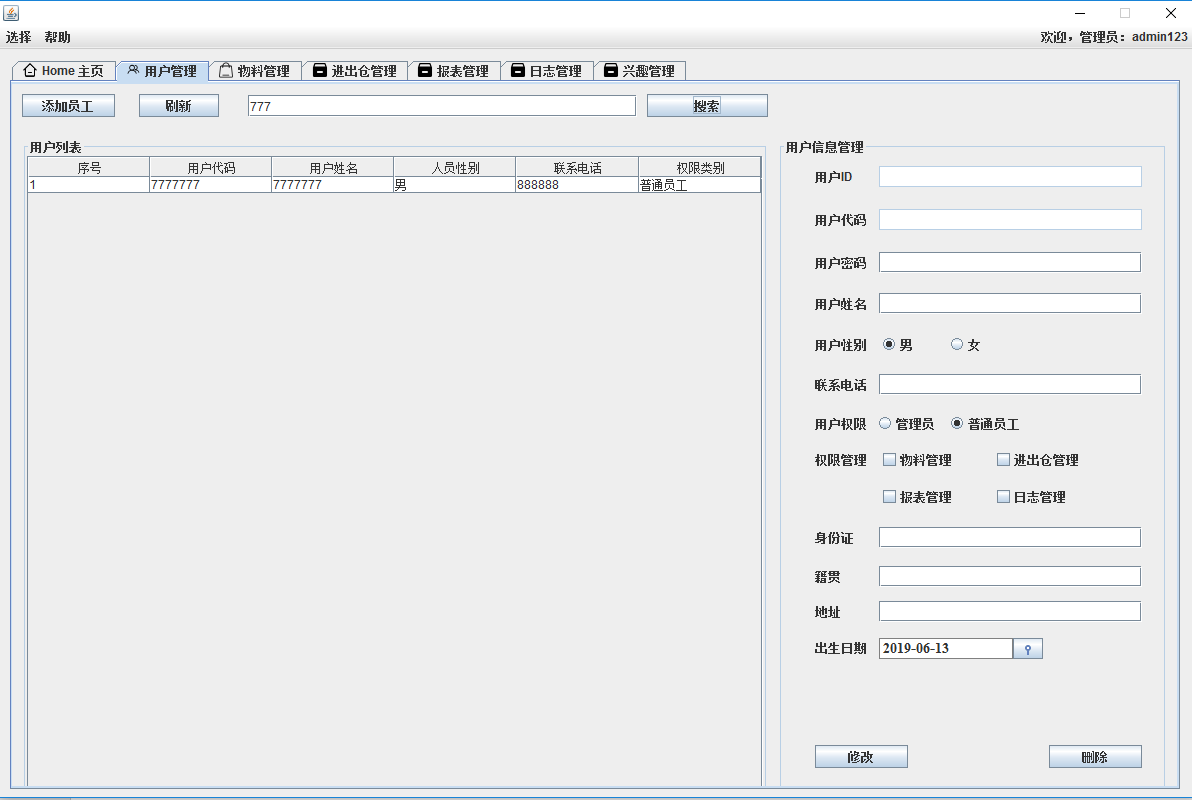


#### 1.5.5 主要测试



### [1.6、人员档案查询](#_Toc12683)

#### 1.6.1 具体界面



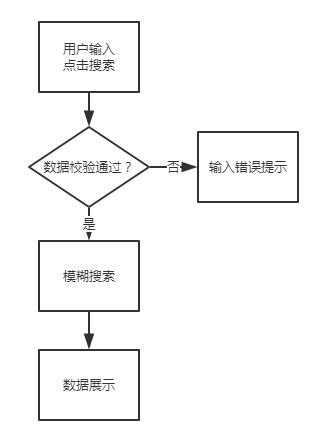
#### 1.6.2 实现思路

1、通过定义的搜索框获取用户输入

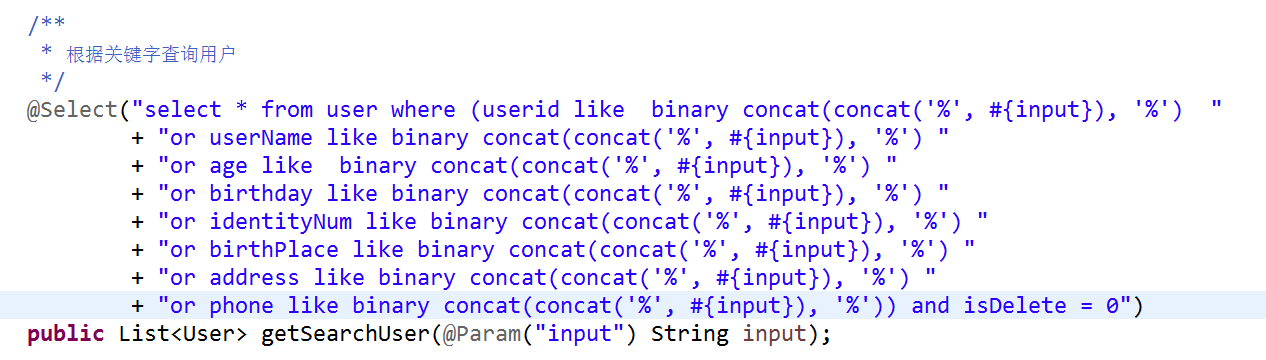
2、使用模糊查询的方式查询符合的所有用户信息返回

3、封装查找到的数据信息，使用表格的方式展示给用户

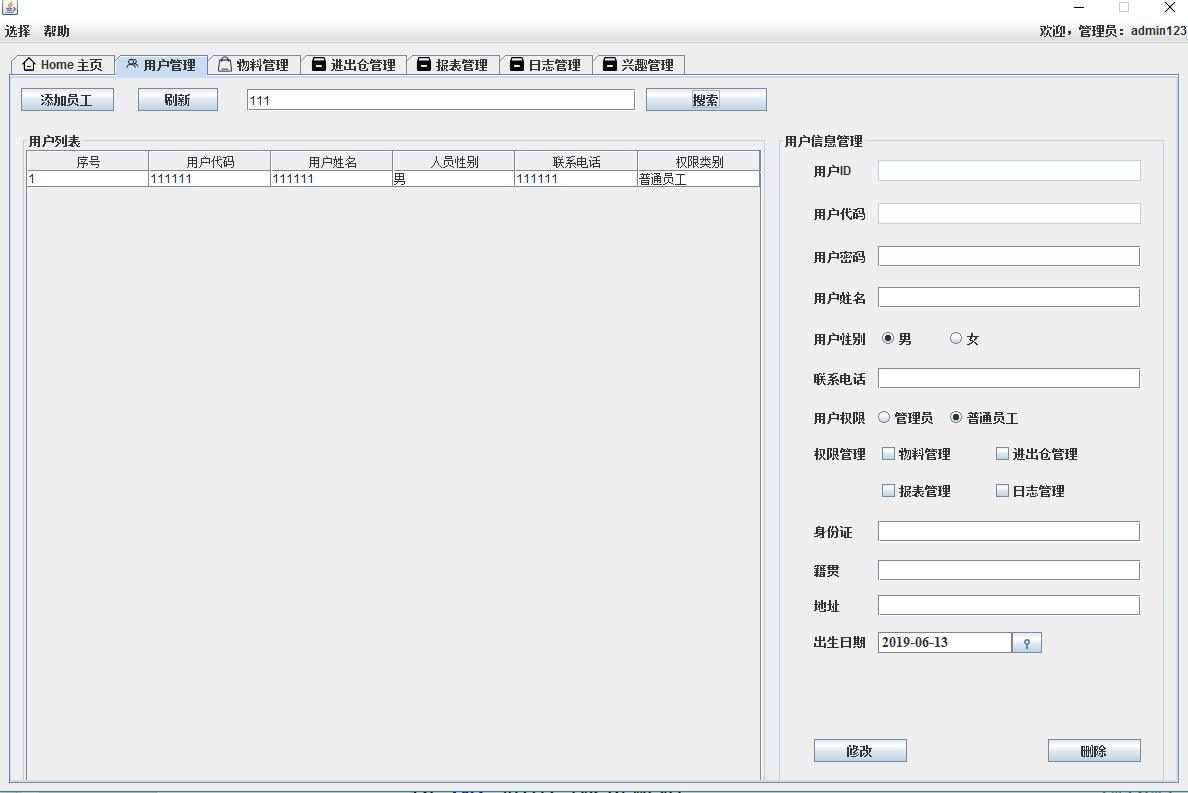
#### 1.6.3 具体流程



#### 1.6.4 主要数据库操作



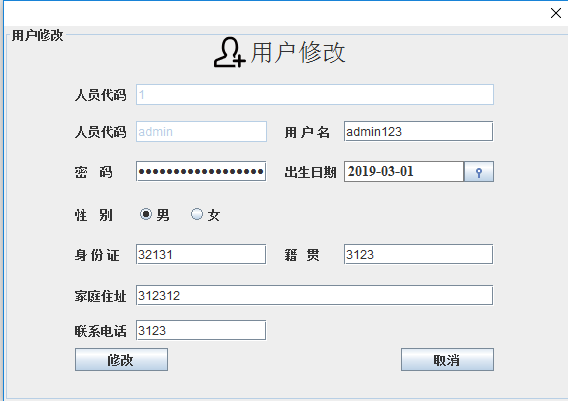
#### 1.6.5 主要测试



### 1.7 其他功能

#### 1.7.1 登录用户修改

##### 1.7.1.1 具体界面



##### 1.7.1.2 实现思路

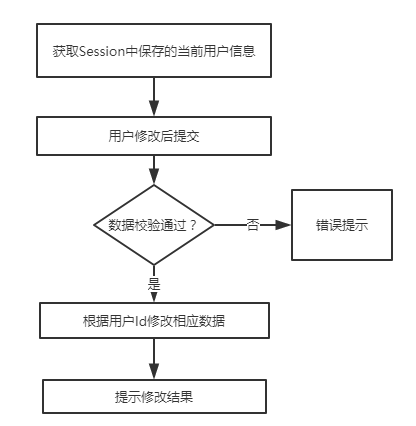
1、登录用户会将信息保存到Session缓存中，通过Session缓存获取到当前登陆用户信息，初始化用户修改界面，将用户数据填写到相应的文本框内。

2、用户修改信息，点击修改提交表单信息，判断密码是否是加密过的，如果密码是加密后（32位）则密码不变，若密码修改，则使用MD5进行加密。

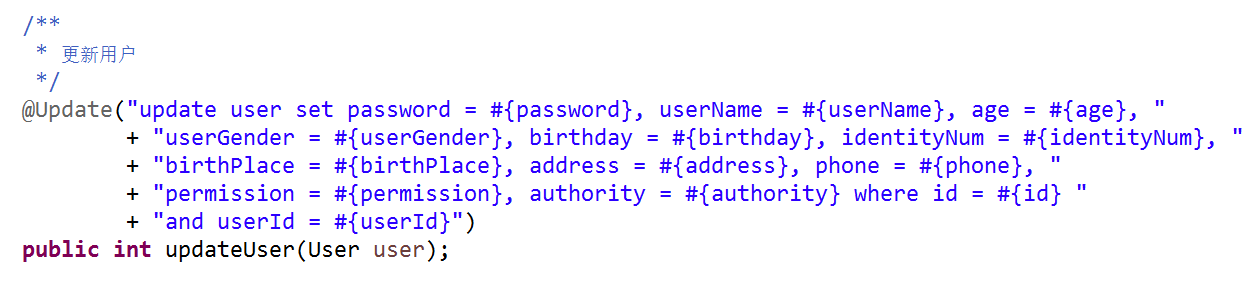
3、后台根据id和用户代码（userId）作为依据修改相应用户记录。

1. 返回相应执行结果，展示提示界面。

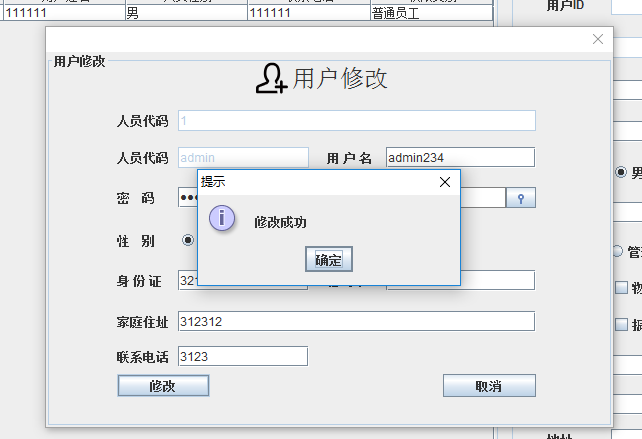
##### 1.7.1.3具体流程



##### 1.7.1.4主要数据库操作



##### 1.7.1.5主要测试



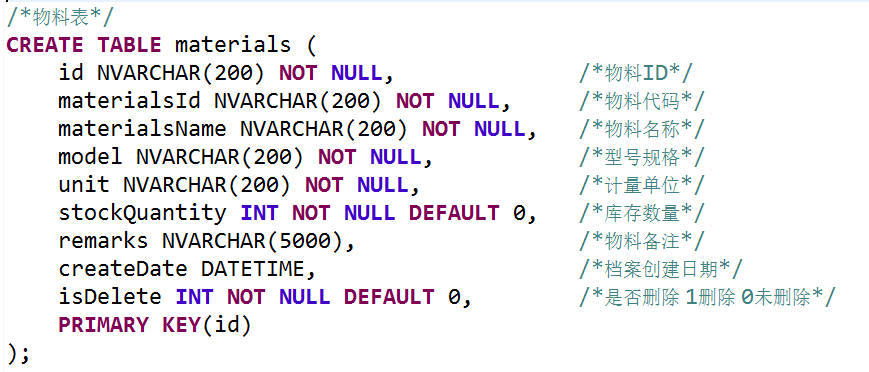
## 2、[物料档案管理](#_Toc28620)

### 2.0、主要需求

物料档案包括物料代码、物料名称、规格型号、计量单位、库存数量、备注等字段。

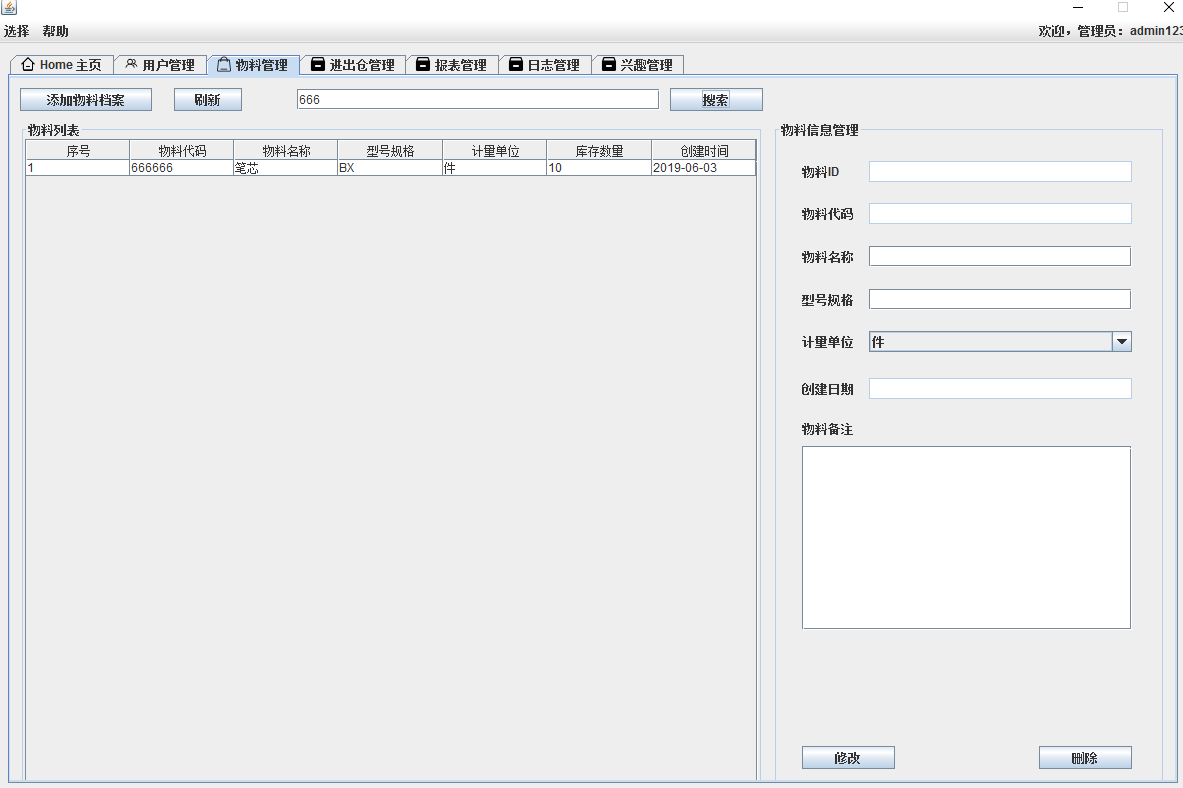
1. 物料档案的增加、修改、删除程序，具体内容参考人员档案管理程序。
2. 客户端输入时要求物料代码不为空，计量单位可以选择“件、套、公斤、吨、升、米、毫米、个”等，计量单位并可以输入其它单位名，库存数量默认为0。
3. 物料查询程序可以在通过输入框的文字进行模糊查询（物料代码、名称、规格中是否含有输入框的文字）。

### 2.1、数据库表设计



### [2.2、物料档案查询](#_Toc25286)

#### 2.2.1 具体界面设计



#### 2.2.2 实现思路

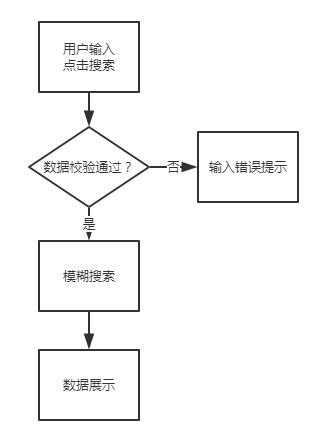
1、通过定义的搜索框获取用户输入

2、使用模糊查询的方式查询符合的所有物料信息返回

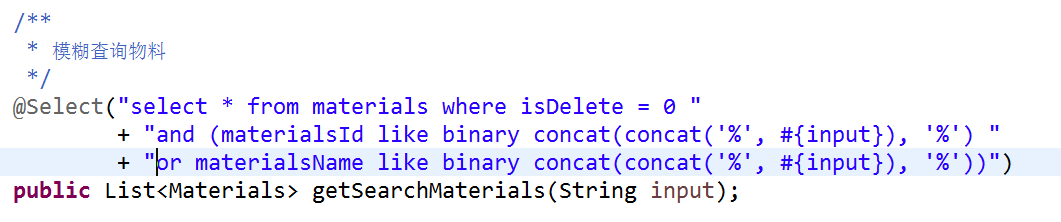
3、封装查找到的数据信息，使用表格的方式展示给用户

可通过相应的物料代码进行模糊查询

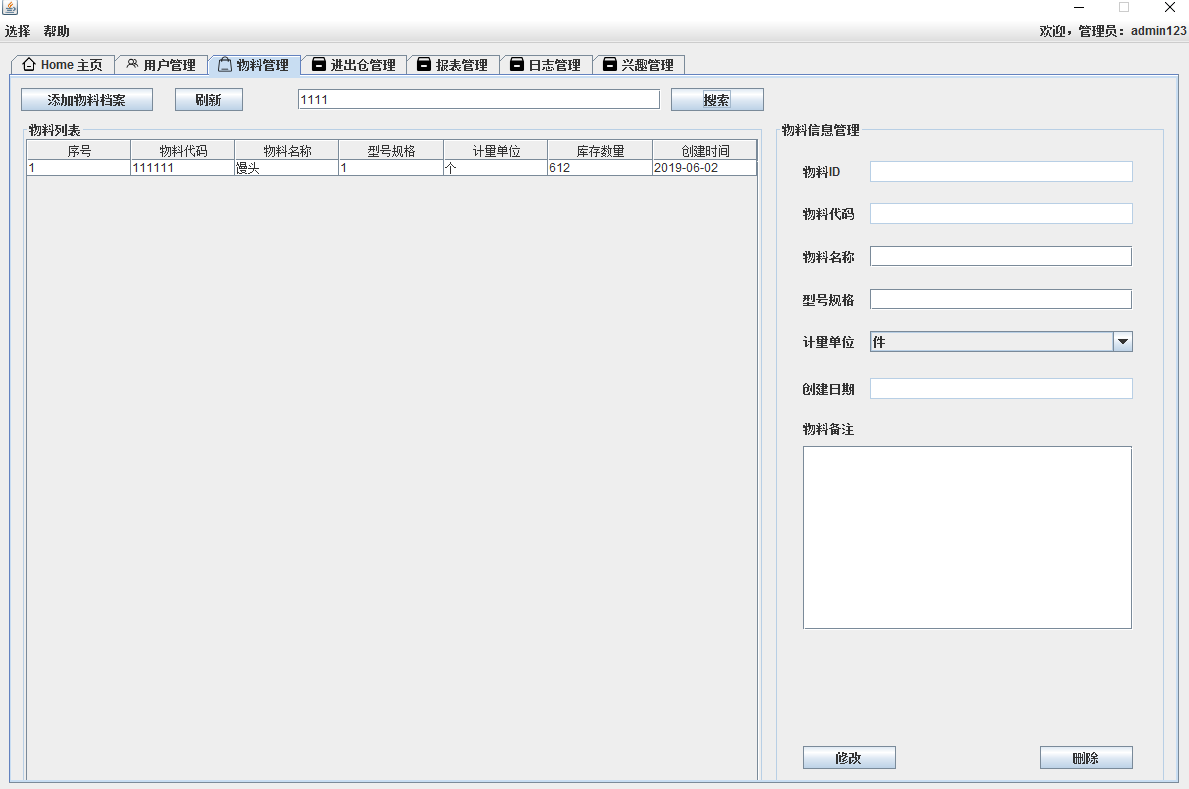
#### 2.2.3 具体流程图



#### 2.2.4 主要数据库操作

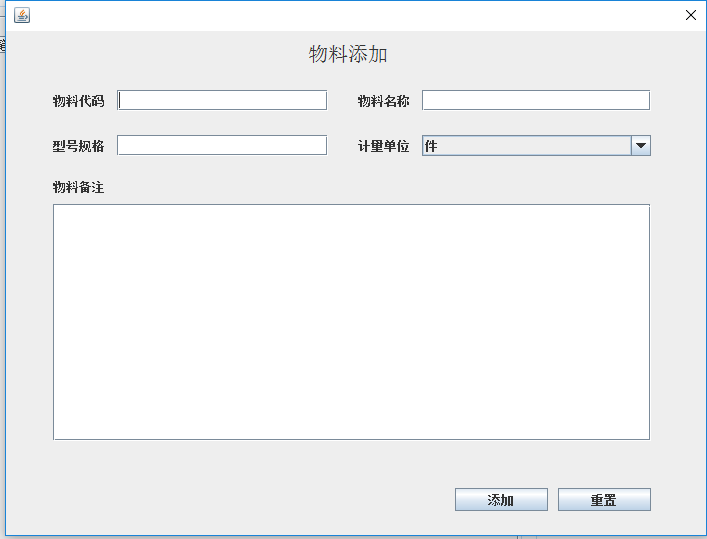


#### 2.2.5 主要测试



### [2.3、物料档案添加](#_Toc25830)

#### 2.3.1 具体界面设计



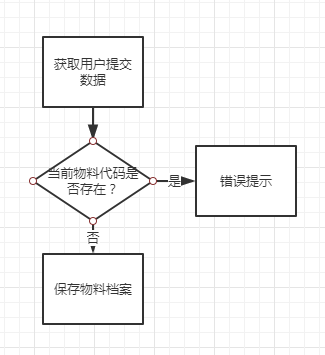
#### 2.3.2 实现思路

1、通过表单获取用户输入的物料信息，判断当前提交的物料代码是否存在，若存在则显示提示界面，终止添加操作，若不存在则向下执行

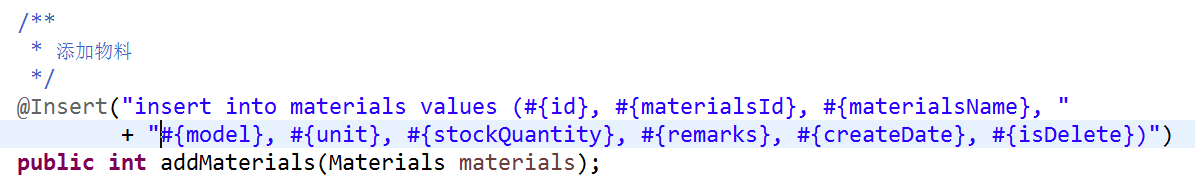
2、向数据库物料档案表中添加当前记录

3、根据添加操作结果返回相应提示信息

#### 2.3.3 具体流程图



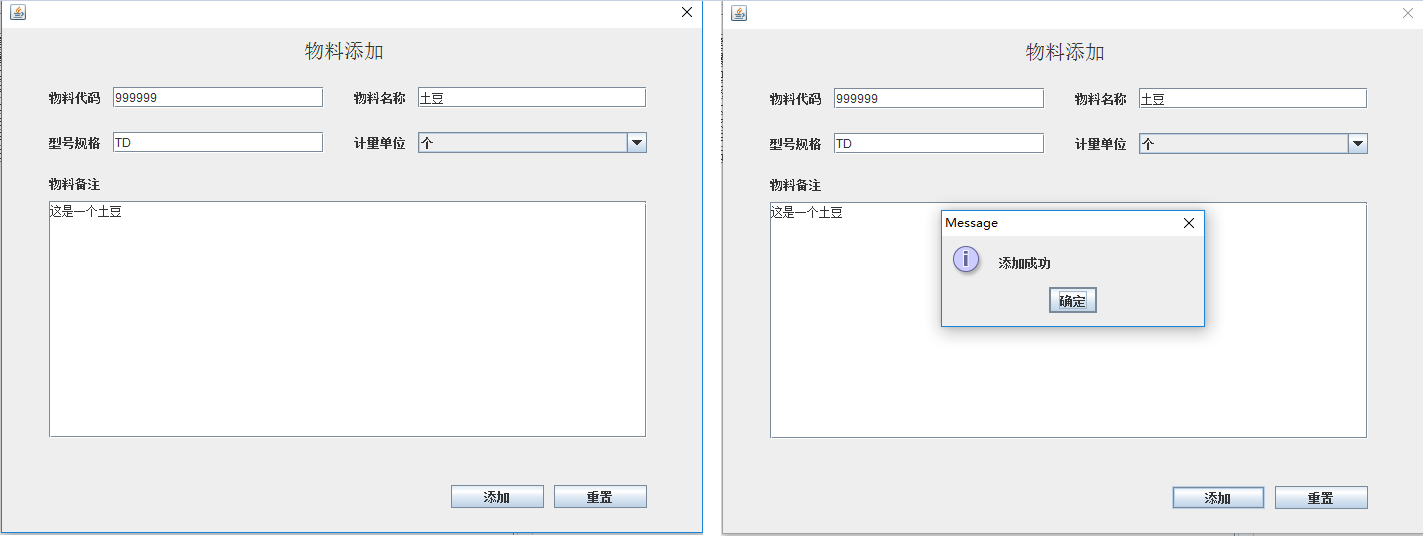
#### 2.3.4 主要数据库操作



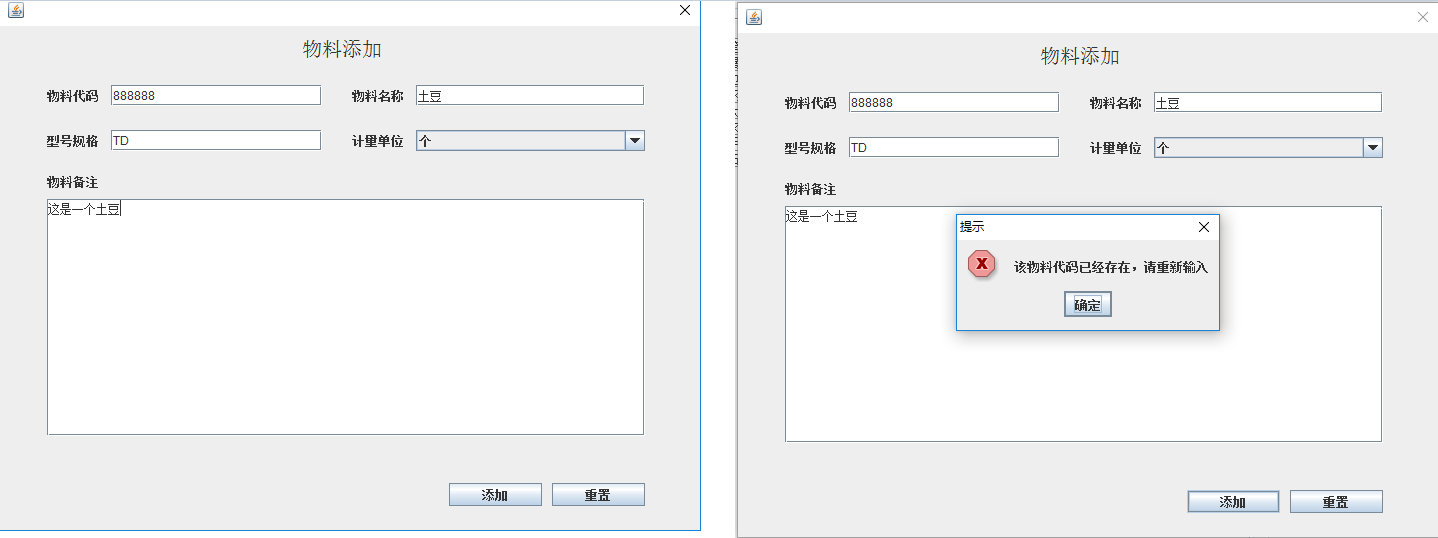
#### 2.3.5 主要测试

测试添加物料档案

成功（物料：土豆）

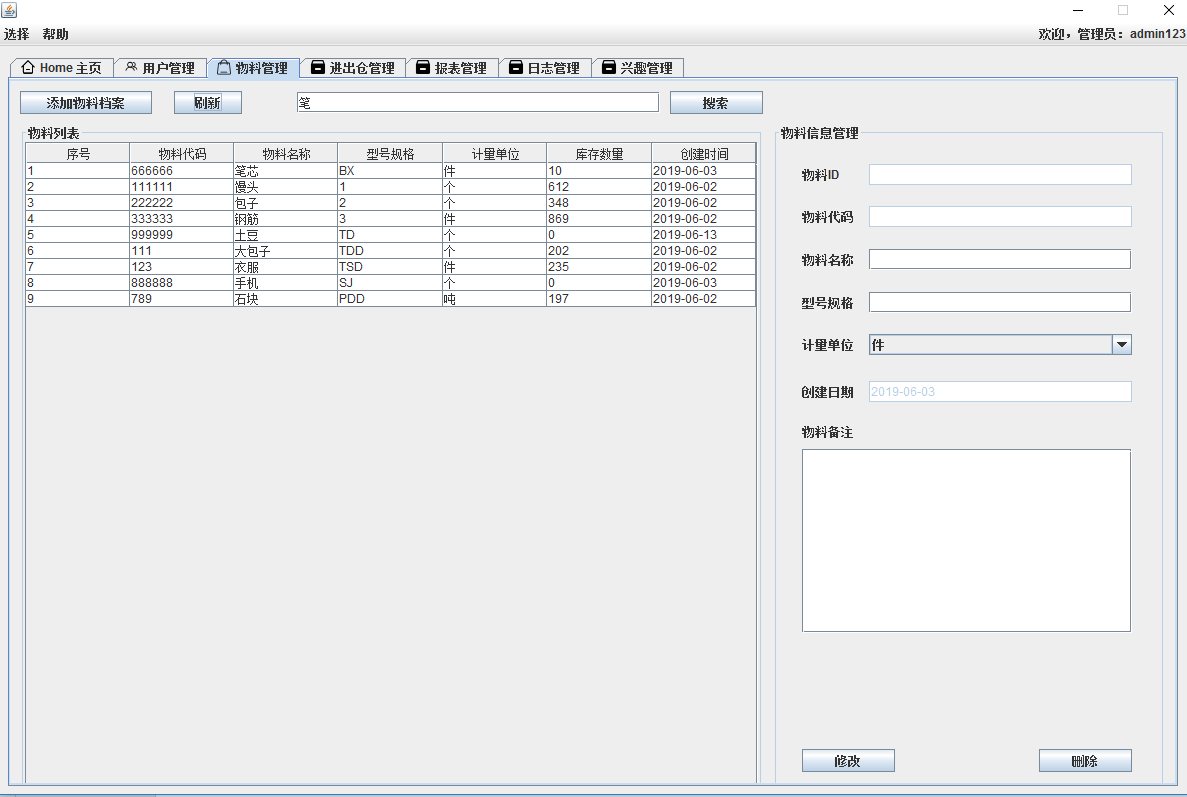


失败（物料代码相同）



### [2.4、物料档案删除](#_Toc29037)

#### 2.4.1 具体界面设计



#### 2.4.2 实现思路

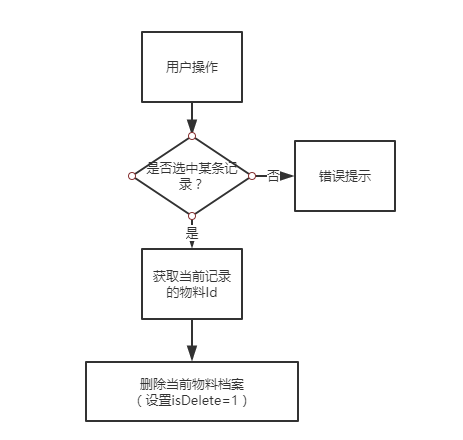
1、当选择某一行时会自动提取当前行的materialsId（物料代码）数据。

2、根据提取到的materialsId，在数据库中查询到相应物料的所有信息，并将查找到的信息填充到上图物料信息管理表单中。

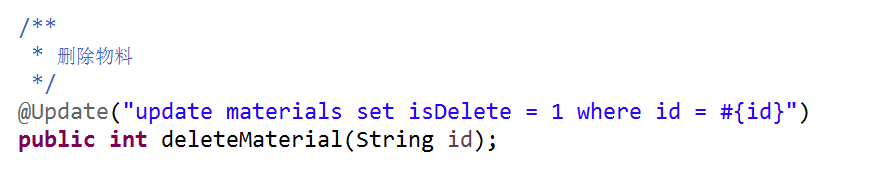
3、当用户点击删除时，会根据id和materialsId删除当前记录

4、根据返回结果，显示提示信息

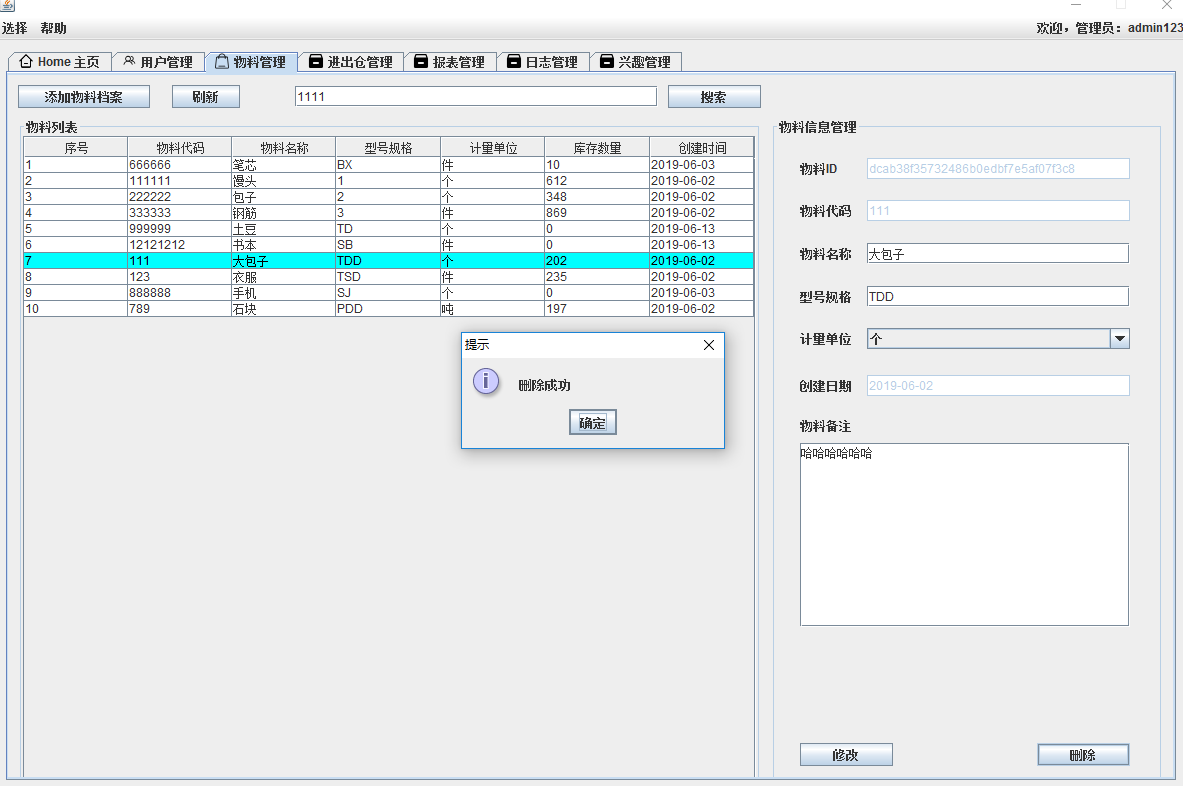
#### 2.4.3 具体流程图



#### 2.4.4 主要数据库操作

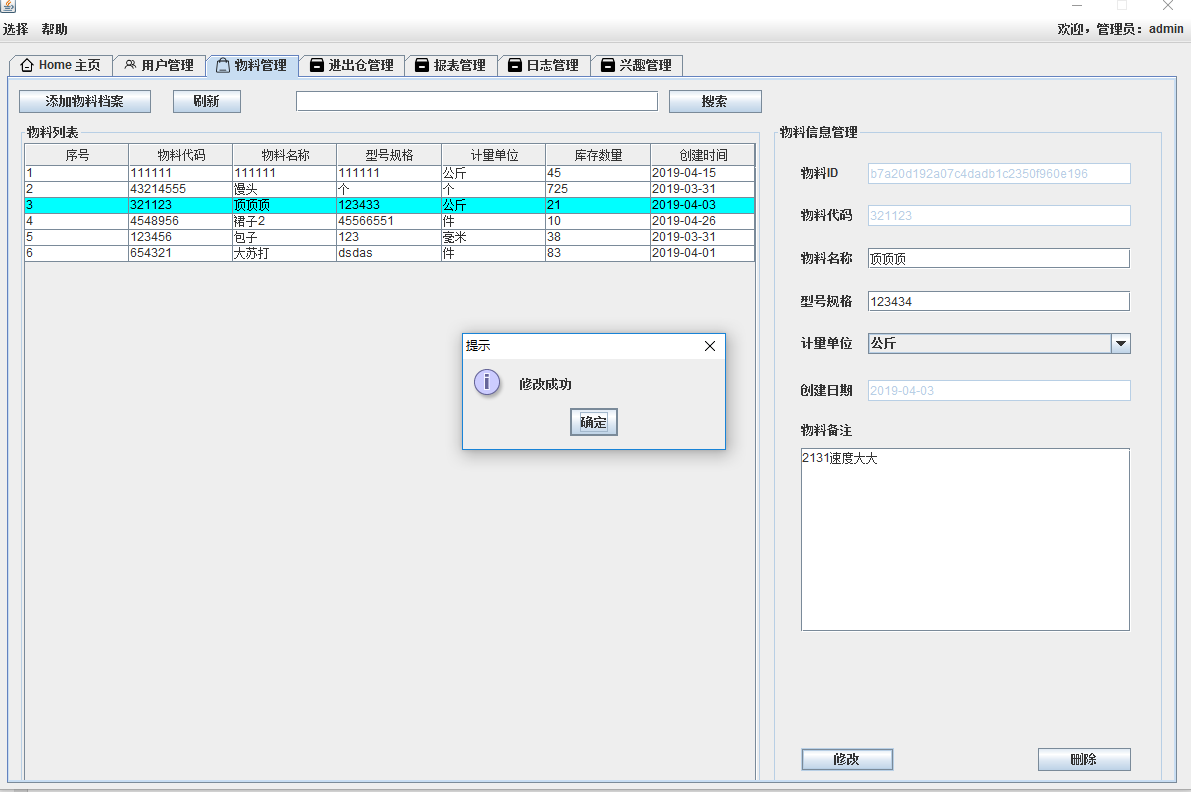


#### 2.4.5 主要测试



### [2.5、物料档案修改](#_Toc23122)

#### 2.5.1 具体界面设计



#### 2.5.2 实现思路

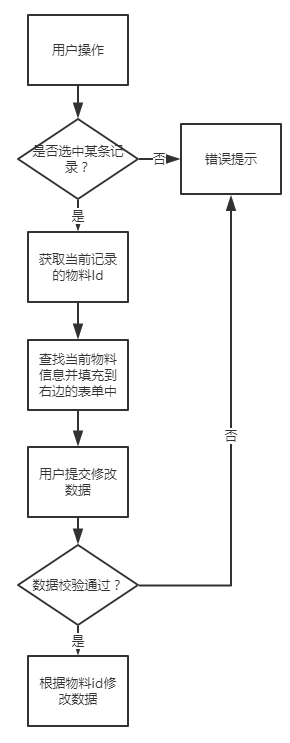
1、当选择某一行时会自动提取当前行的materialsId（物料代码）数据。

2、根据提取到的materialsId，在数据库中查询到相应物料的所有信息，并将查找到的信息填充到上图物料信息管理表单中。

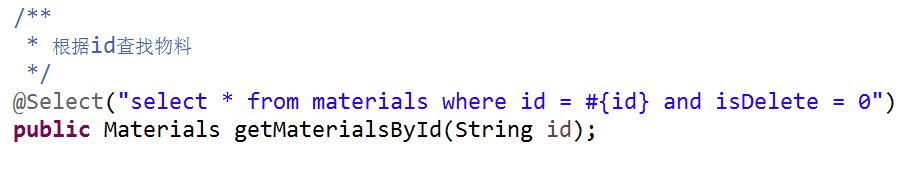
3、提交表单，获取到提交的表单数据后，根据id和物料代码（materialsId）作为依据修改相应物料记录。

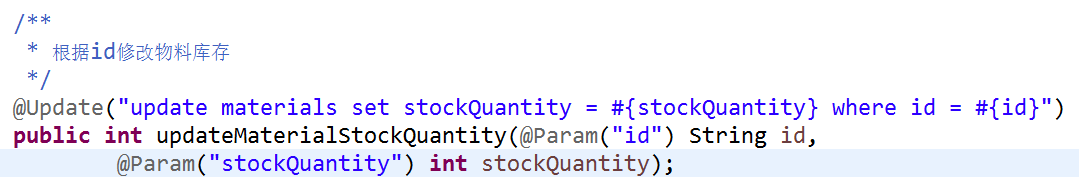
4、返回相应执行结果，展示提示界面。

#### 2.5.3 具体流程图

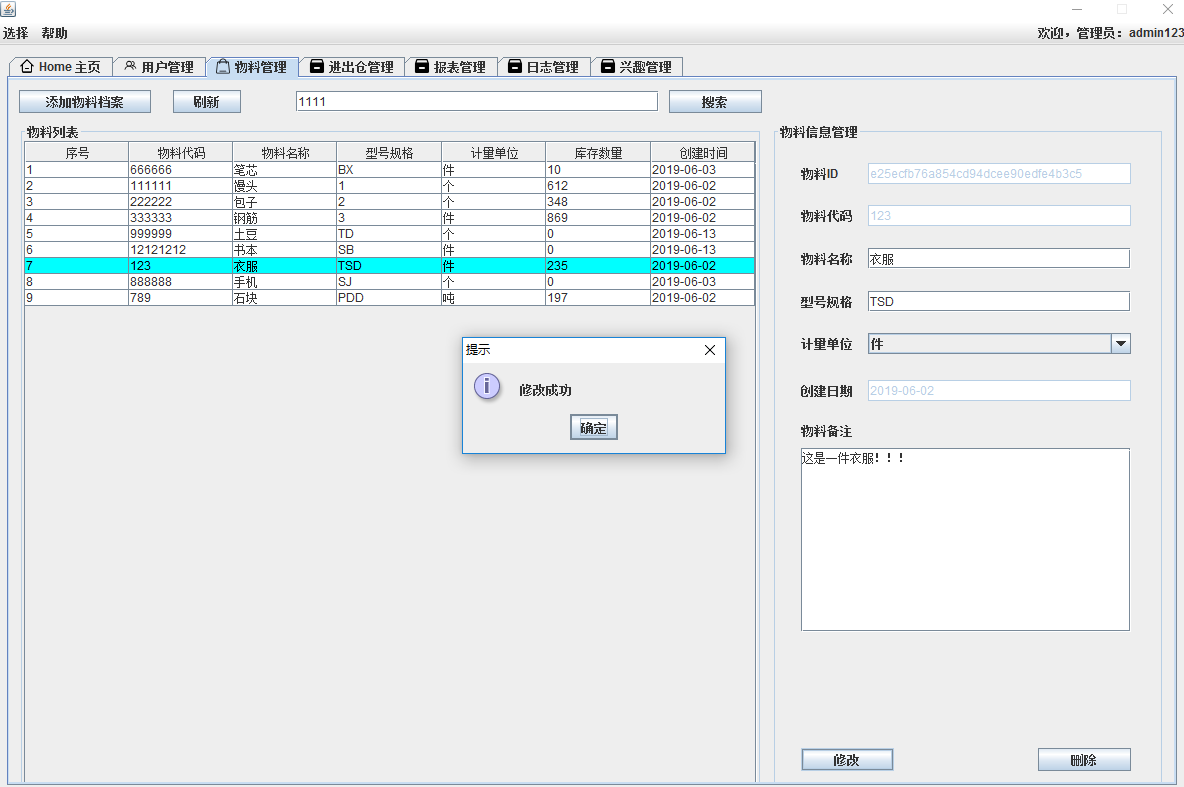


#### 2.5.4 主要数据库操作





#### 2.5.5 主要测试



# 二、[实验课题二，课题三：单/多物料进出仓信息管理](#_Toc28421)

## 1、进出仓管理

### 主要需求

**实验课题二、简单物料进仓/出仓信息管理**

企业物料进出仓的信息管理，物料进出仓的信息主要包括进出仓单号、进出仓日期、操作人员代码、备注、物料代码、进仓数量、出仓数量等

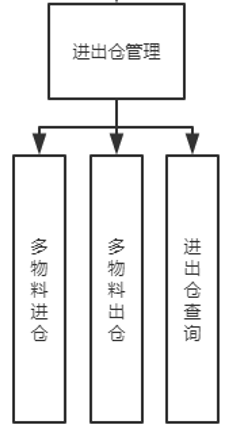
1. 企业仓库进出仓信息表，可以用一个表，也可以用二个表（一对多关系），具体表设计的思路须在报告中描述。
2. 进仓程序与出仓程序可以合并，也可以分开做。其中操作人员必须用下拉框来选择；单号要自己编规则自动生成，单号产生规则在报告中说明；进出仓日期用日期控件；物料代码必须下拉框做选择；当物料出仓时必须判断出仓数量时候超过物料的库存数量，物料的库存数量在物料表中可以获得该物料的库存数量。进行进仓或出仓操作时必须通过调用存储过程来实现，存储过程将操作结果返回调用者，显示成功或失败的信息。
3. 进仓、出仓的存储过程可以合并或分开做，存储过程要判断物料是否存在，操作后物料库存数量是否不为负数；如果不满足条件显示，失败提示；如果满足条件，增加进出仓表记录一条，并根据进出仓物料的数量对物料表中库存数量进行修改，提交数据库操作，否则回滚。
4. 进出仓单查询程序，可以用进出仓日期起止条件，物料代码、操作人员、备注（用子串查询）来完成。

**实验课题三、多物料进出仓信息管理**

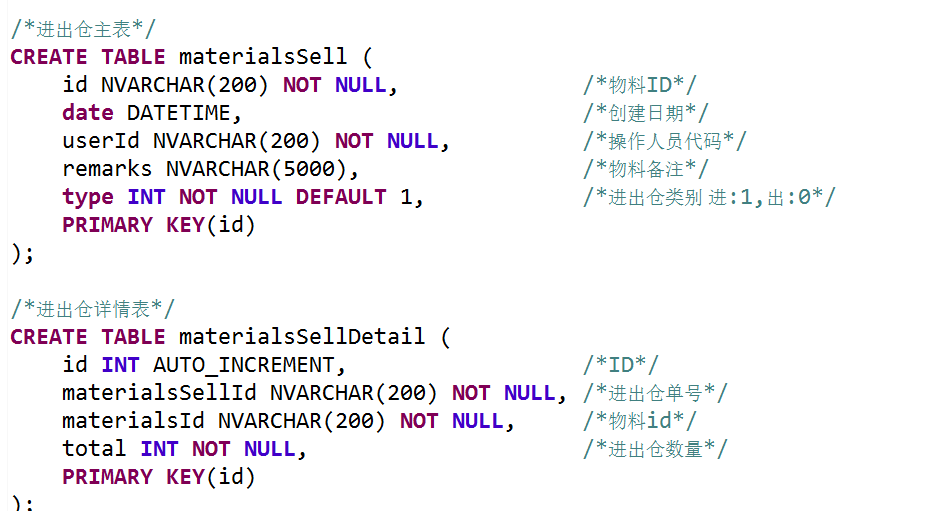
一般企业进行物料进出仓时，多在一个进出仓单号上同时对多种物料做同进或同出操作，通过程序实现多个物料在一个界面一个单号上同时实现进仓或出仓操作，如果不成功，必须全部回滚。

1. 多物料进仓出仓，可以在一个程序中完成，也可以分开编程，同一单号中进出仓单号、进出仓日期、操作人员必须完全相同，同一单号中不能进仓与出仓同时存在，通过调用存储过程完成进出仓操作。
2. 多物料进出仓存储过程，多物料进出仓单增加到表中，并修改多个物料的库存数量，失败就显示提示内容。

根据实验二与实验三的要求内容可以看出两者的本质是相近的，都是对物料进行进出仓操作，故而在我们讨论之后决定将两题做成一题，不独立出单独的模块，使整个应用的功能设计更为合理。

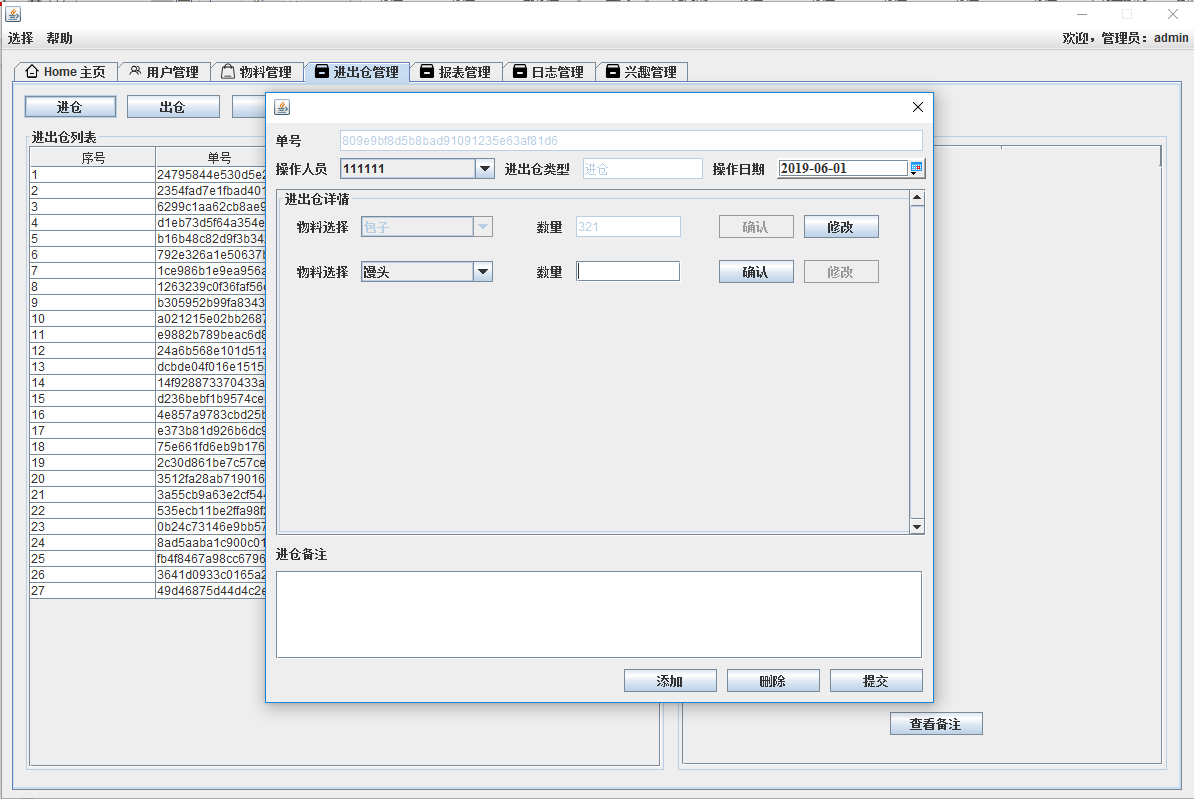


### 1.1数据库表设计



### 1.2物料进仓

#### 1.2.1 具体界面设计



#### 1.2.2 实现思路

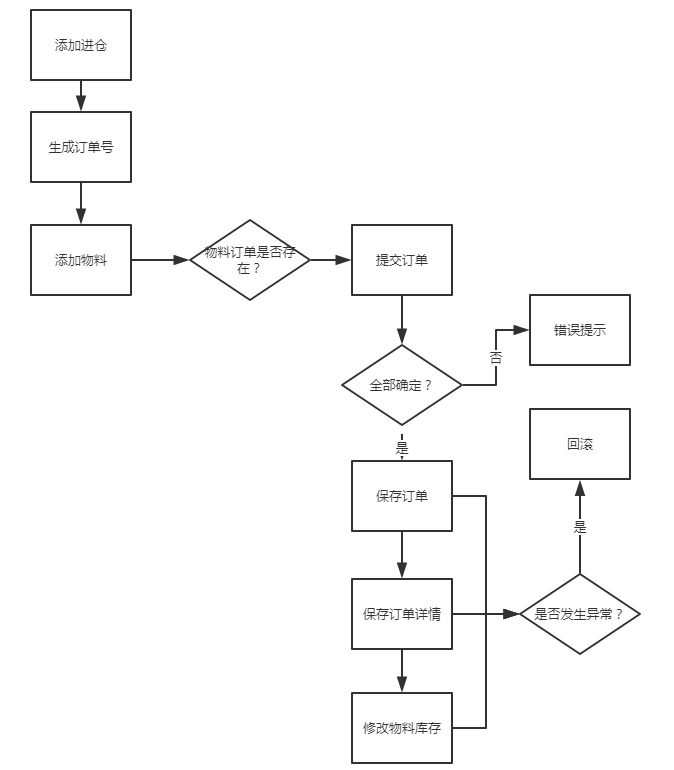
1.当用户点击进仓按钮时查询数据库内所有的用户信息和物料档案信息（用于生成下拉列表），自动生成一个32位的随机订单号。

2.点击添加按钮添加相应的详情订单，点击确认按钮后，将该详情订单信息保存到一个Set集合中，保证添加的详情订单中不会有重复的物料订单。

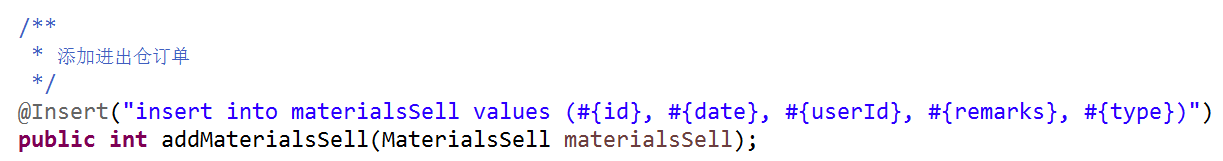
3.点击提交会提交两组数据，一组是订单数据，一组是订单详情数据，开启Spring事务，向数据库中同时保存两组数据，在保存数据的同时需要修改物料档案中的库存数量，若三组数据库操作都成功则向下。若操作过程中出现错误，则抛出异常，Spring根据捕获的Exception异常回滚数据。

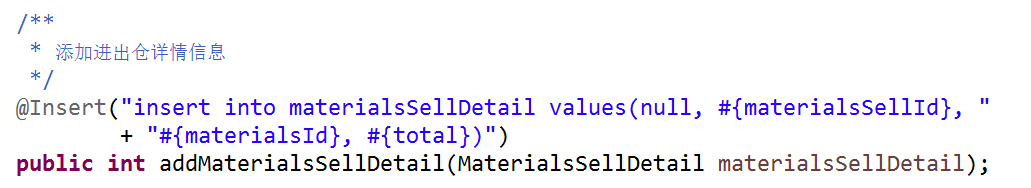
4.返回操作结果，显示提示信息。

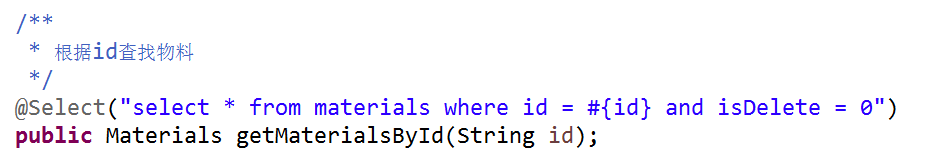
#### 1.2.3 具体流程图

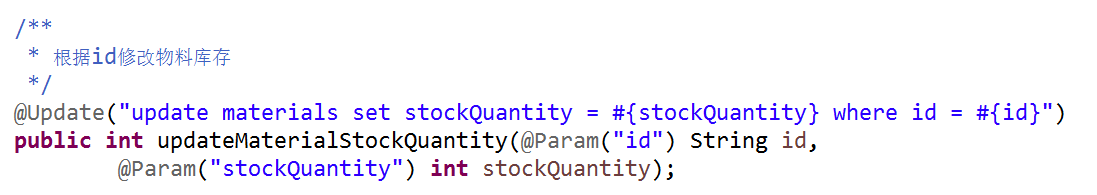


#### 1.2.4 主要数据库操作



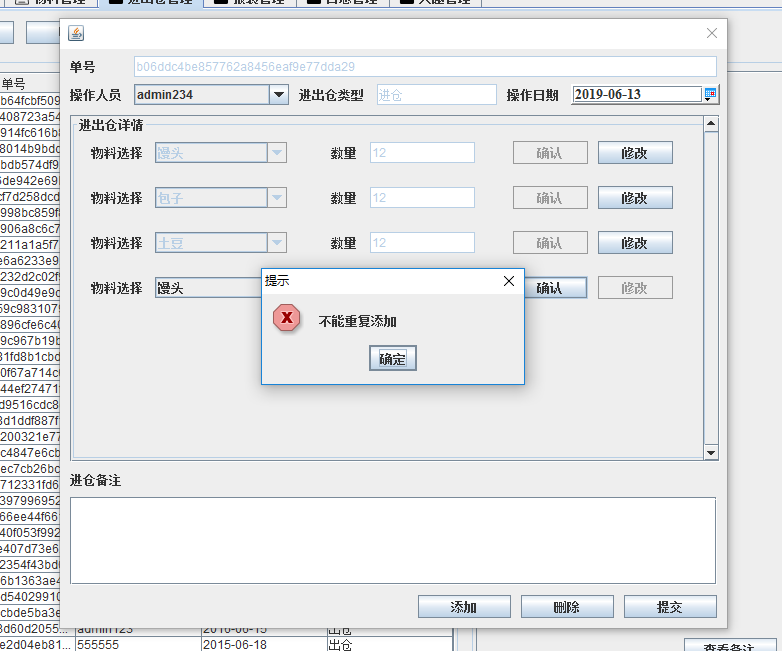




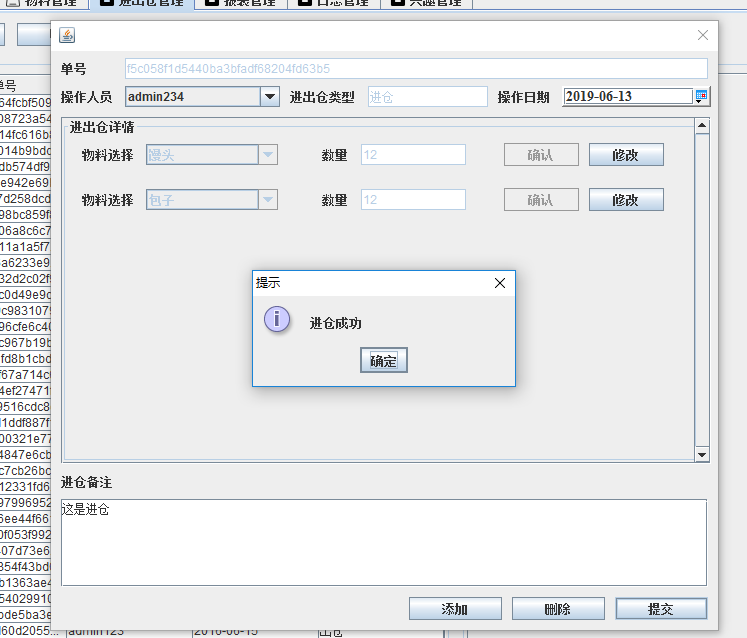


#### 1.2.5 主要测试

不能重复添加

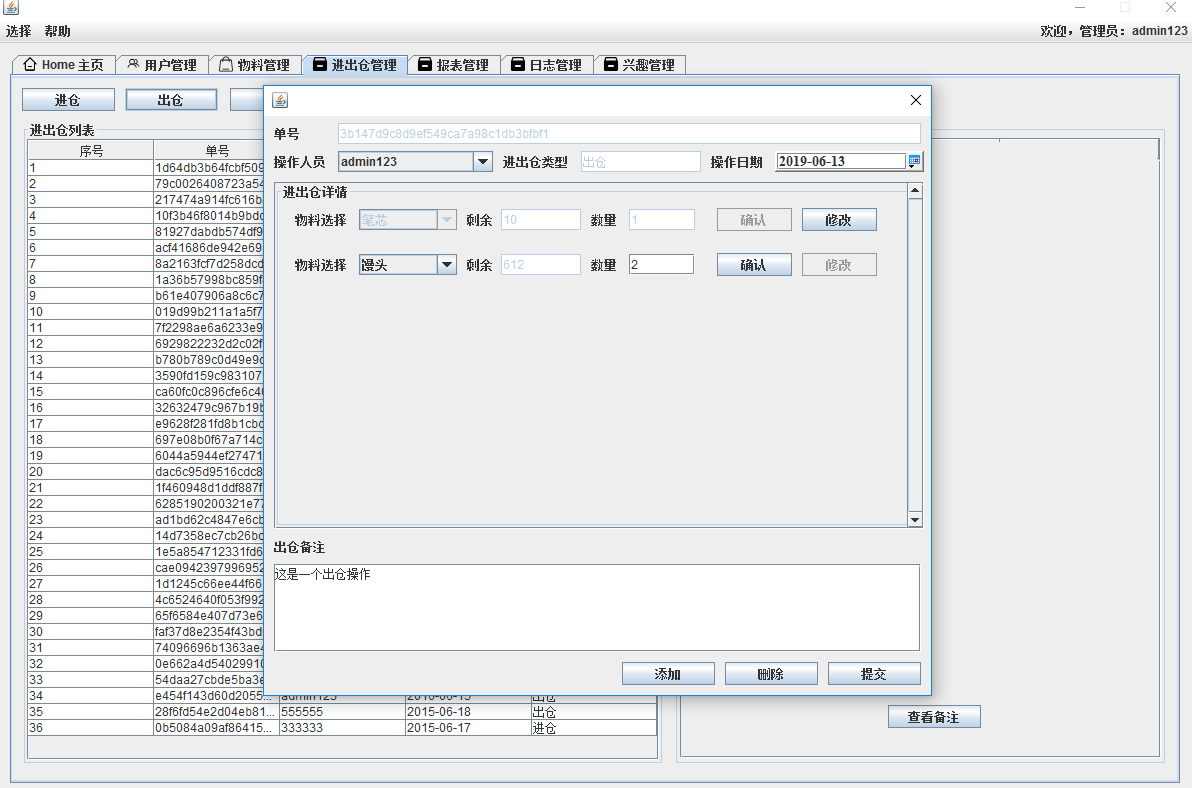


成功：



### 1.3 物料出仓

#### 1.3.1 具体界面设计



#### 1.3.2 实现思路

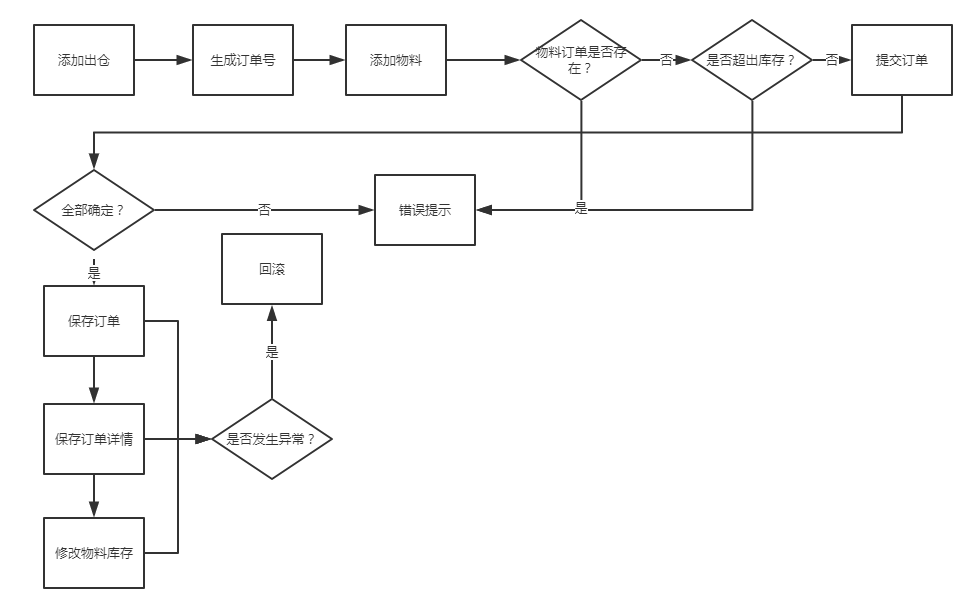
1.当用户点击出仓按钮时查询数据库内所有的用户信息和物料档案信息（用于生成下拉列表），自动生成一个32位的随机订单号。

2.点击添加按钮添加相应的详情订单，点击确认按钮后，将该详情订单信息保存到一个Set集合中，保证添加的详情订单中不会有重复的物料订单。

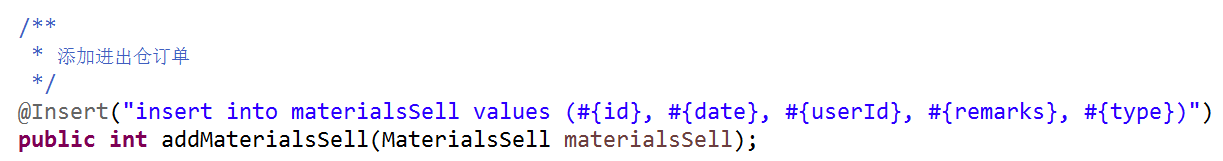
3.点击提交会提交两组数据，一组是订单数据，一组是订单详情数据，开启Spring事务，向数据库中同时保存两组数据，在保存数据的同时需要修改物料档案中的库存数量，若三组数据库操作都成功则向下。若操作过程中出现错误，则抛出异常，Spring根据捕获的Exception异常回滚数据。

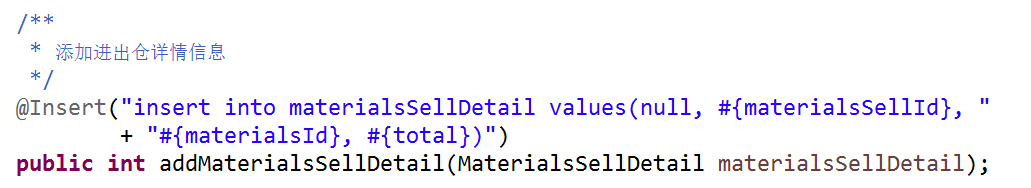
4.返回操作结果，显示提示信息。

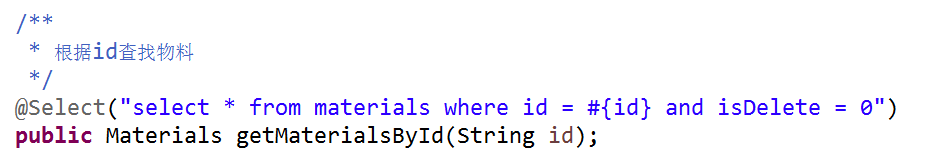
#### 1.3.3 具体流程图

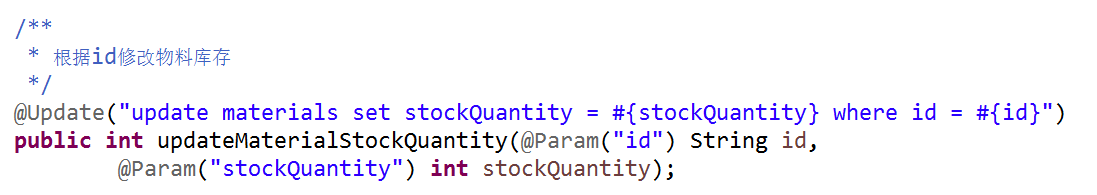


#### 1.3.4 主要数据库操作



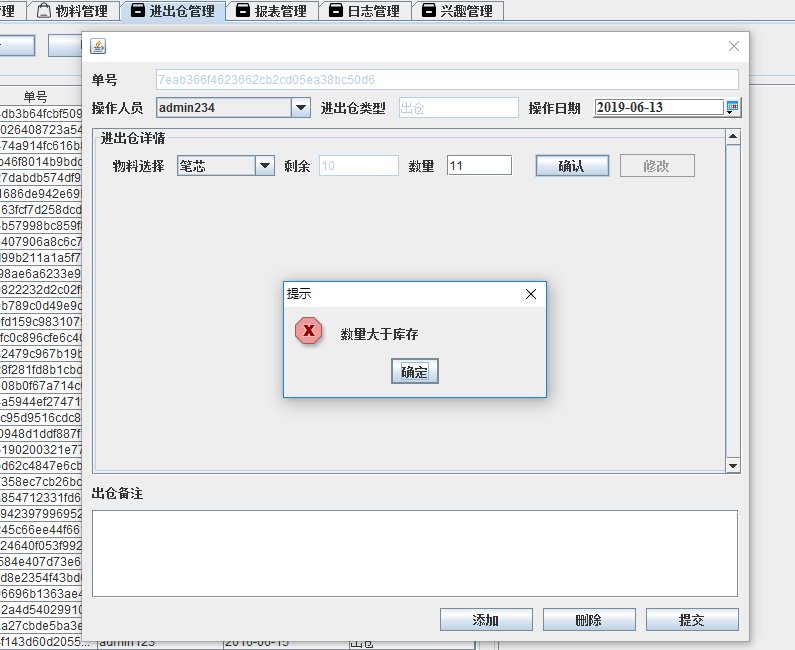




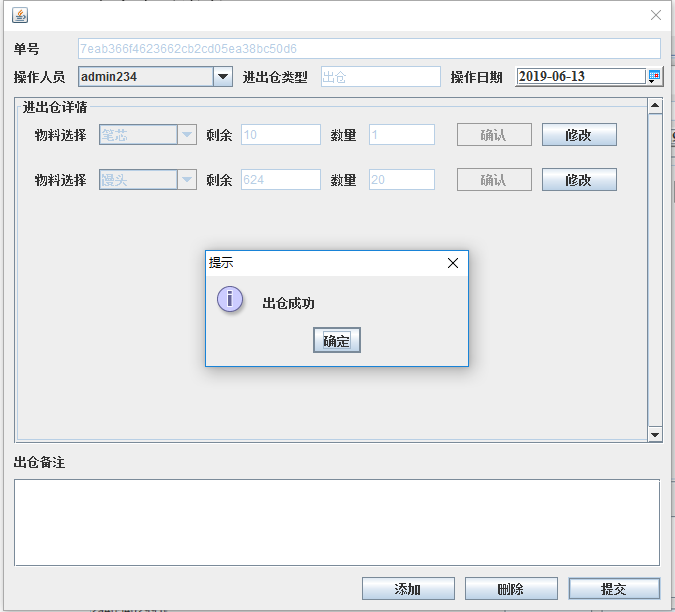


#### 1.3.5 主要测试

出仓大于库存

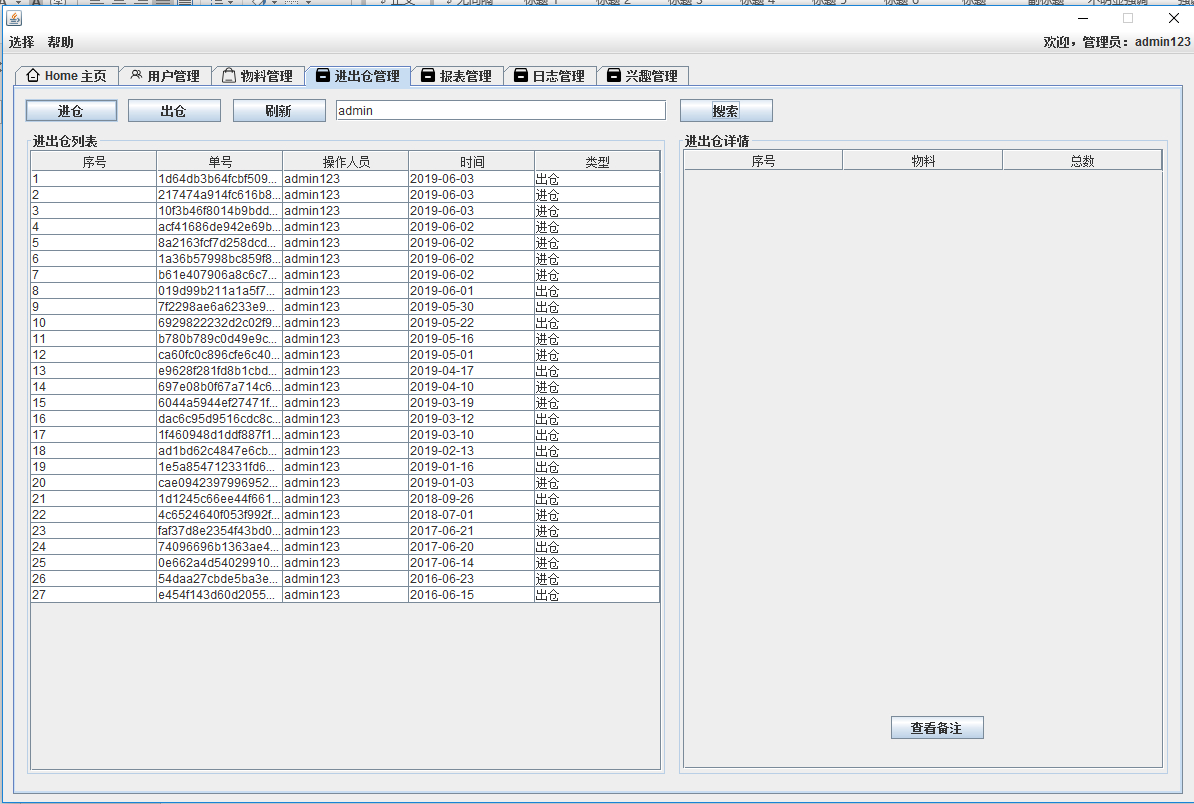


成功：



### 1.4 物料订单搜索

#### 1.4.1 具体界面设计



#### 1.4.2 实现思路

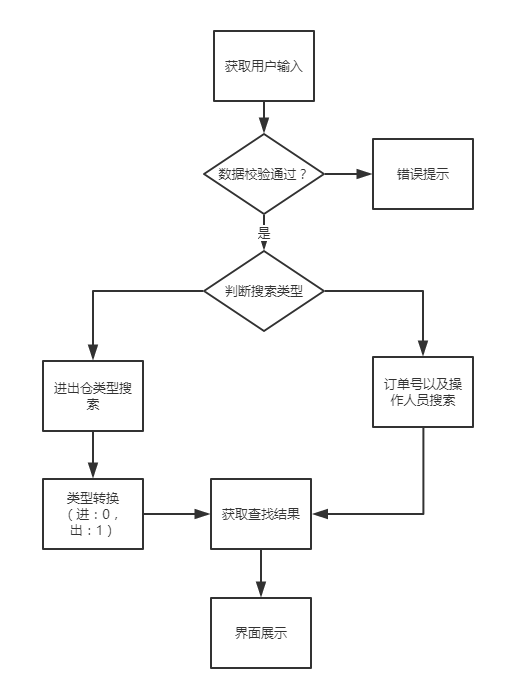
1、通过定义的搜索框获取用户输入

2、使用模糊查询的方式查询符合的所有用户信息返回

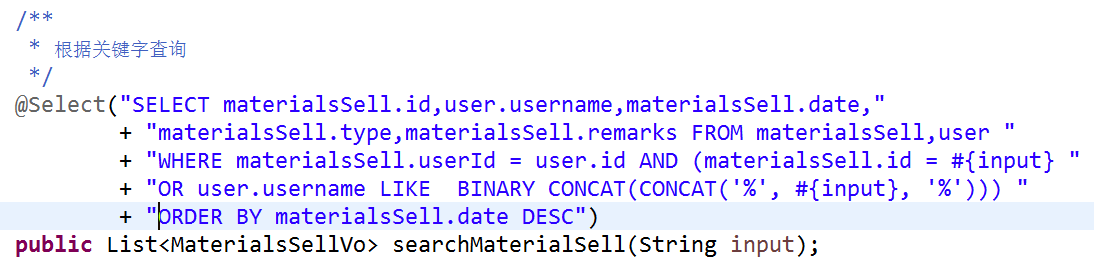
3、封装查找到的数据信息，使用表格的方式展示给用户

可以进行操作人员模糊搜索，订单号搜索，进出仓类型搜索

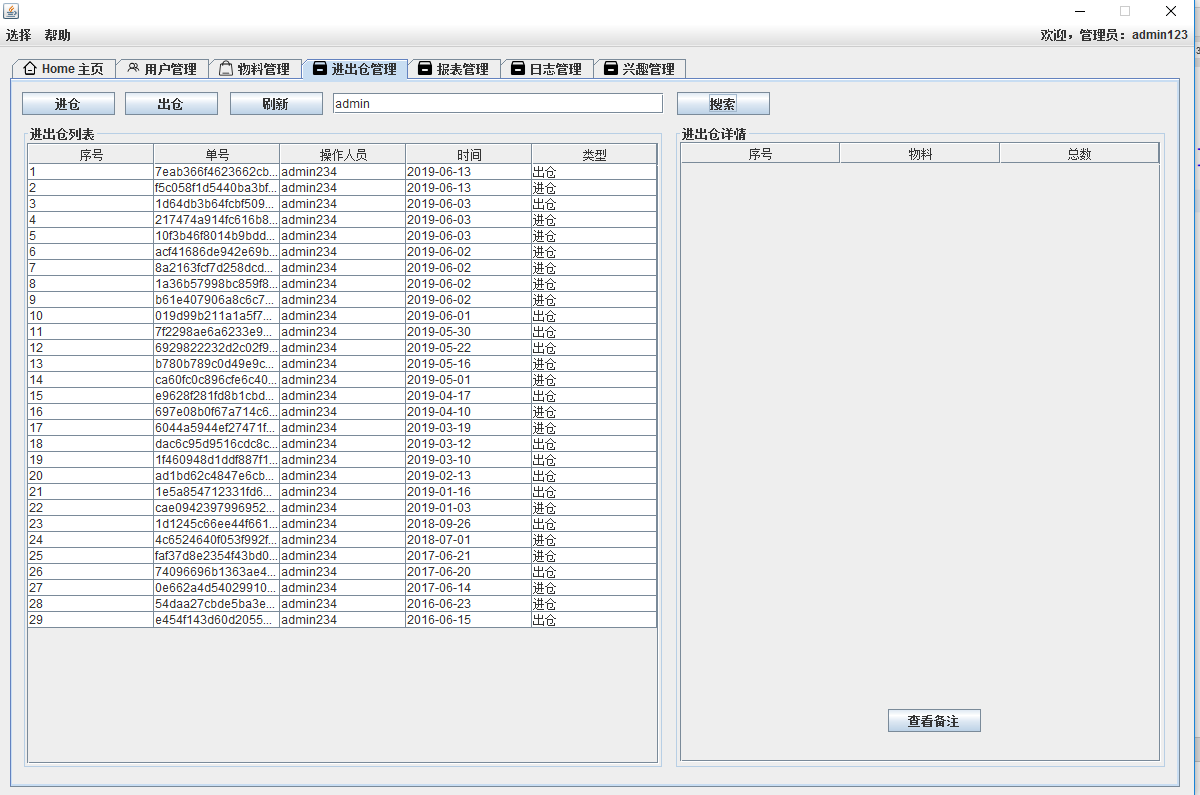
#### 1.4.3 具体流程图



#### 1.4.4 主要数据库操作



#### 1.4.5 主要测试



# 三、[实验课题四：用户登录与权限设计](#_Toc30911)

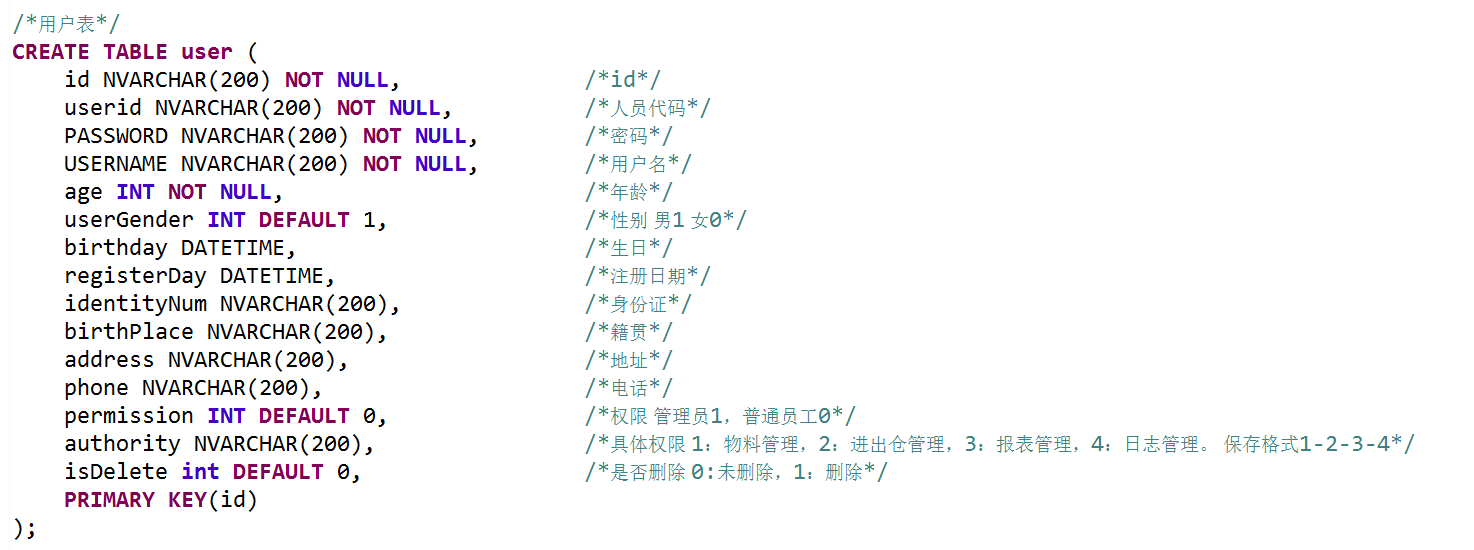
## 1.0 主要需求

**实验课题四、用户登录与权限设计**

用户在这里就指操作人员，已经在人员档案中进行管理，通过设计用户登录和用户权限的管理，实现用户按权限访问资源，管理员按要求授予用户权限，登录程序安全稳定，设计用户菜单及程序调用。人员表中需要新增字段口令、用户权限。系统中每个操作人员都可以作为用户，当一个用户权限修改时，不会影响其他用户；可以对每个用户单独授予某个程序的执行权，也可以收回其执行权，而不影响其他用户的权限；

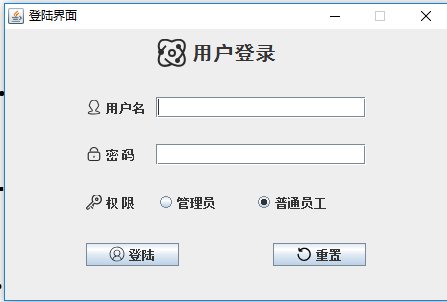
1. 用户登录程序，输入用户名和口令，判断正确后，按用户的权限，显示有权的菜单项，无权菜单项必须不可见或不可操作。
2. 用户授权程序。有授权权限的用户，可以指定某个用户获取某个程序执行权，或被禁止某个程序的执行权。

## 1.1 数据库表设计



## 1.2 用户登录

#### 1.2.1 具体界面



#### 1.2.2 实现思路

1、通过界面获取用户名、密码、登陆权限

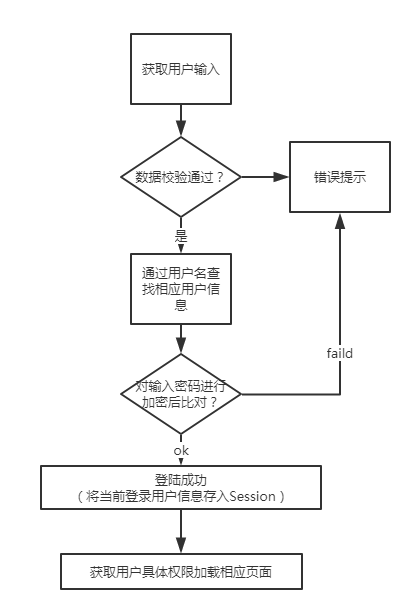
2、根据用户名和权限在数据库中查询相应用户记录，如果有查询到记录，将相应的密码与输入的密码匹配，3、若相符合则用户登陆成功，将用户信息保存到Session缓存中，并关闭登陆界面，打开主页面。

4、若查询不到用户或者用户输入密码与记录密码不符合，则登录失败，提示消息给用户。

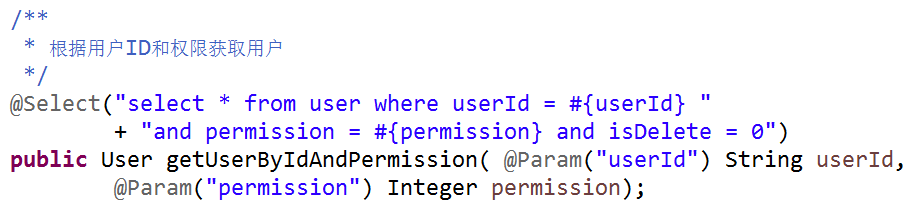
5、用户登陆成功后需要将用户信息展示在主页面右上角，展示当前登陆用户信息。

6、获取用户具体权限加载具体界面

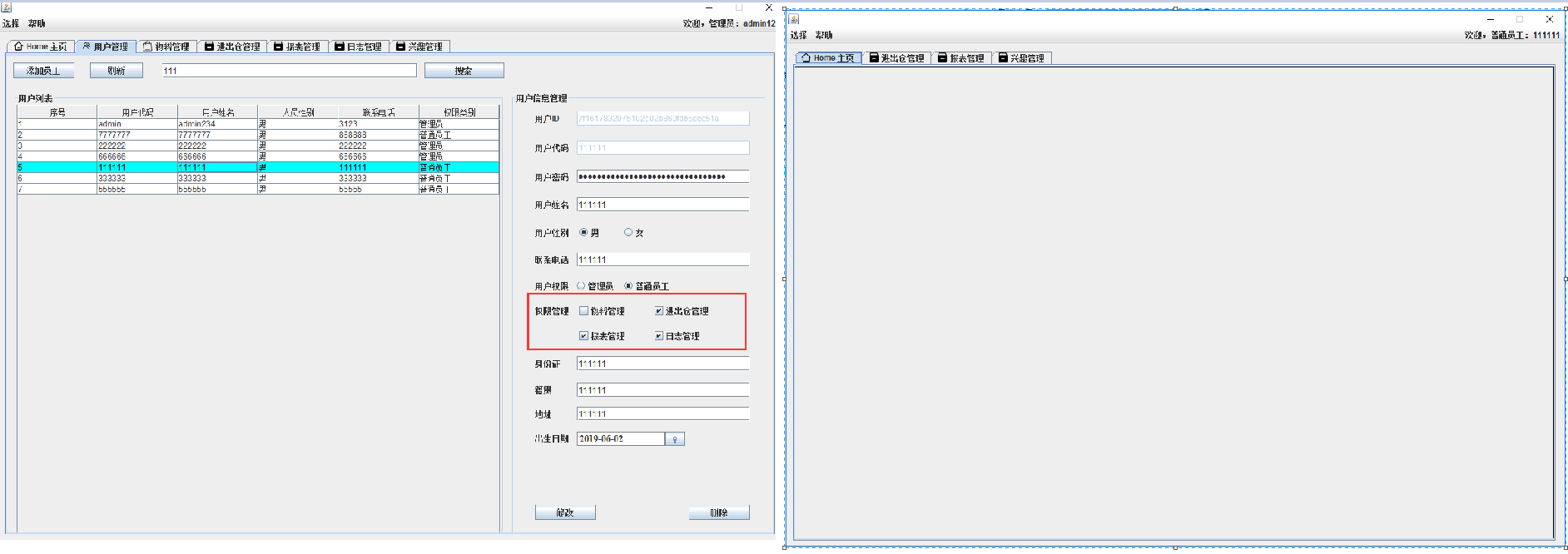
#### 1.2.3 具体流程



#### 1.2.4 主要数据库操作

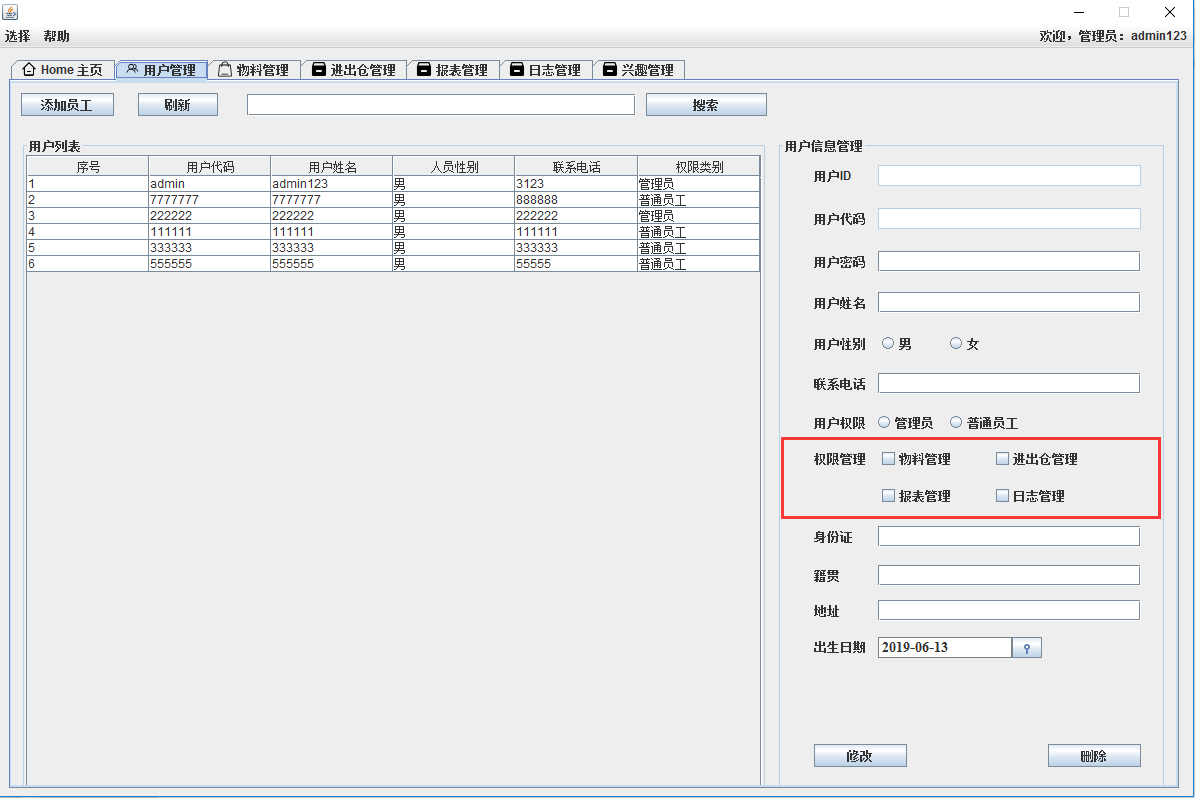


#### 1.2.5 主要测试



## 1.3 权限设计

#### 1.3.1 具体界面

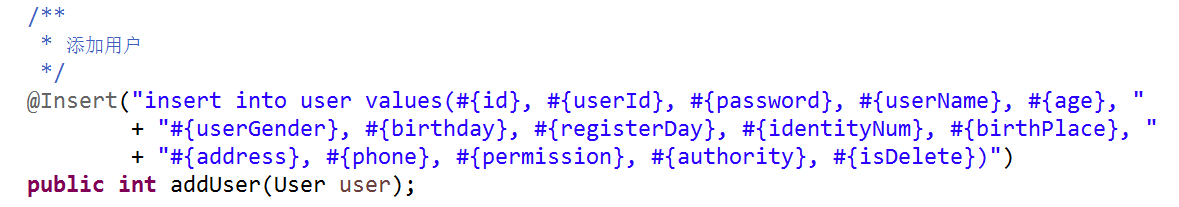


#### 1.3.2 实现思路

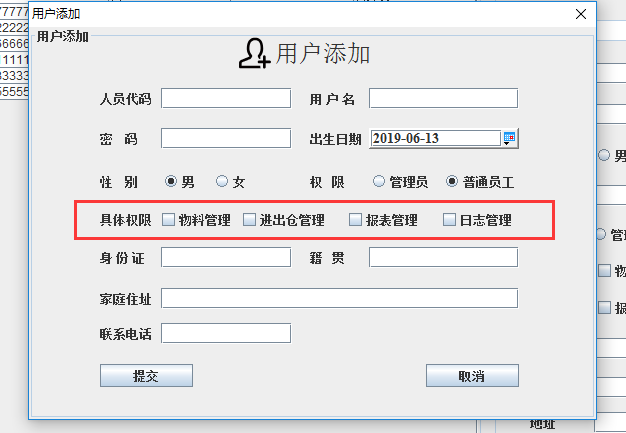
1、获取用户输入，对具体权限设置具体数字，拼接成相应的数据格式保存到数据库 1-2-3

2、当用户登录时会获取相应权限字符串，并解析成单个字符串权限数组，加载相应页面

#### 1.3.3 主要数据库操作



#### 1.3.4 主要测试



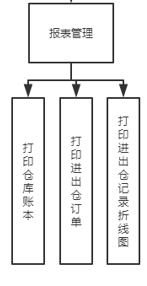
# 四、[实验课题五：统计程序设计与报表打印](#_Toc24242)

## 1.0 主要需求

**实验课题五、统计程序设计与报表打印**

按物料统计进出仓流量，计算分析流动量最小的物料。按月份打印进出仓单表，打印进出仓单，按物料打印仓库账本。

1. 物料统计程序，统计各物料在指定时间内的进出仓数量总数，用图形显示
2. 打印进出仓单程序，给定月份，将该月进出仓单全部按单号顺序打印出来
3. 打印仓库账本程序，给定年份和物料，打印物料代码、名称、规格、计量单位、各日期的进仓、出仓和库存量； EXCEL

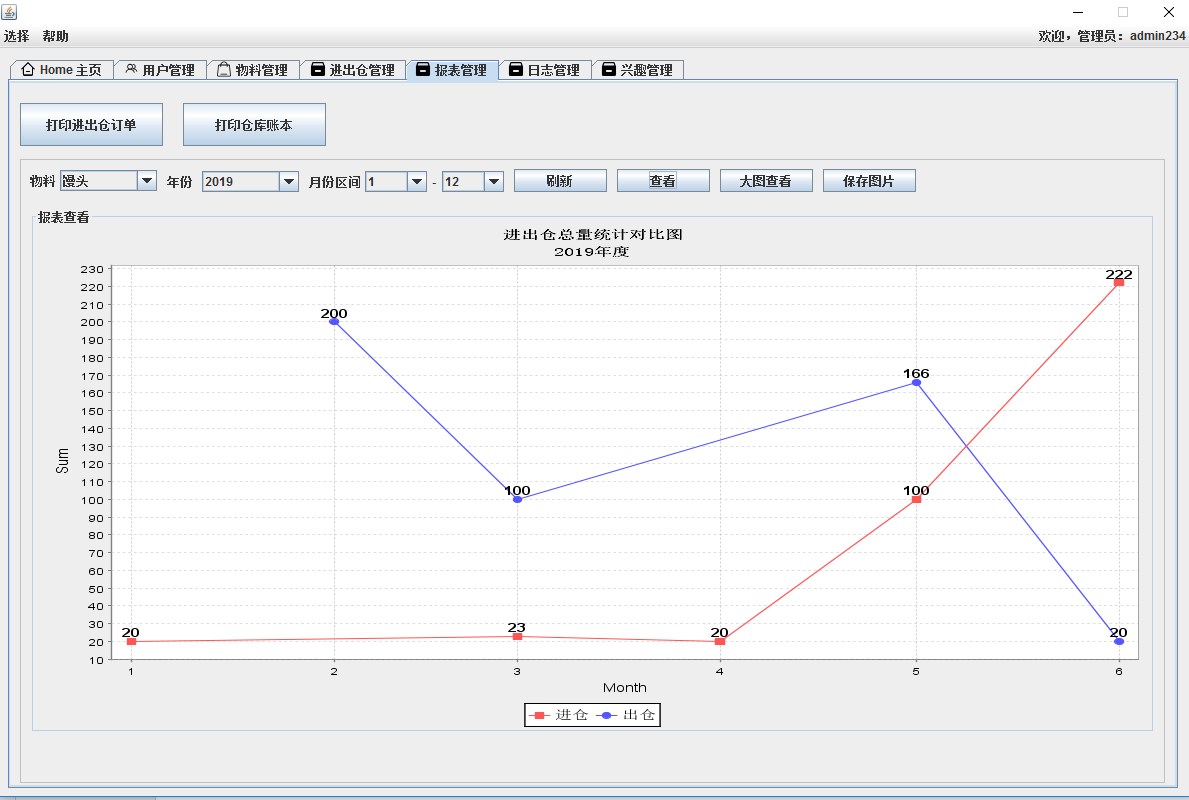


## 1.1 数据库表设计

无

## 1.2 展示某个物料在某个时间段内进出仓折线图

#### 1.2.1 具体界面



#### 1.2.2 实现思路

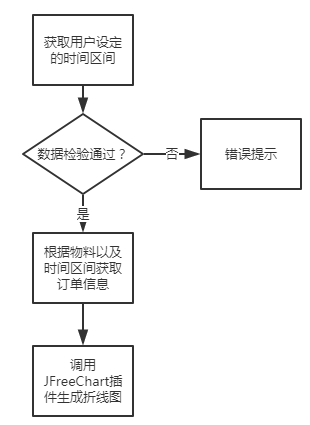
1、获取用户选择的物料以及相应的时间区间

2、查询数据库获取相应的进出仓订单信息，将进仓与出仓记录分开

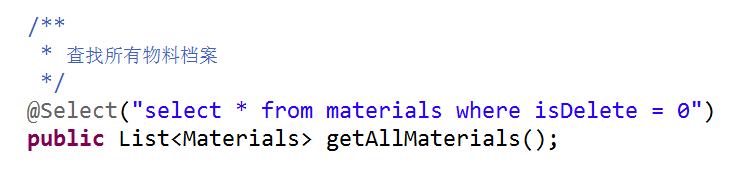
3、将查询的数据封装成具体的数据格式

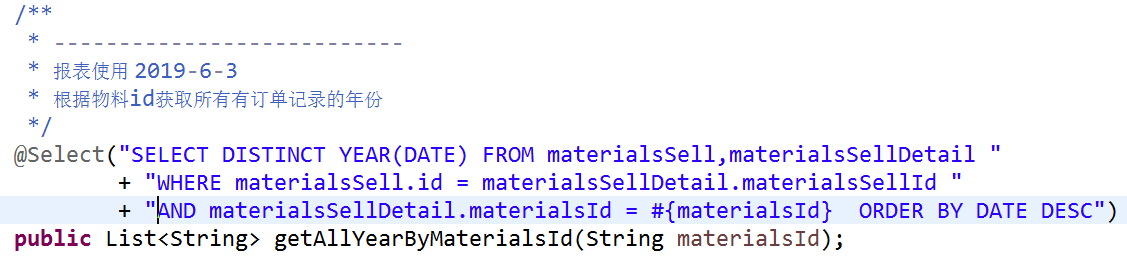
4、调用JFreechart插件生成折线图展示

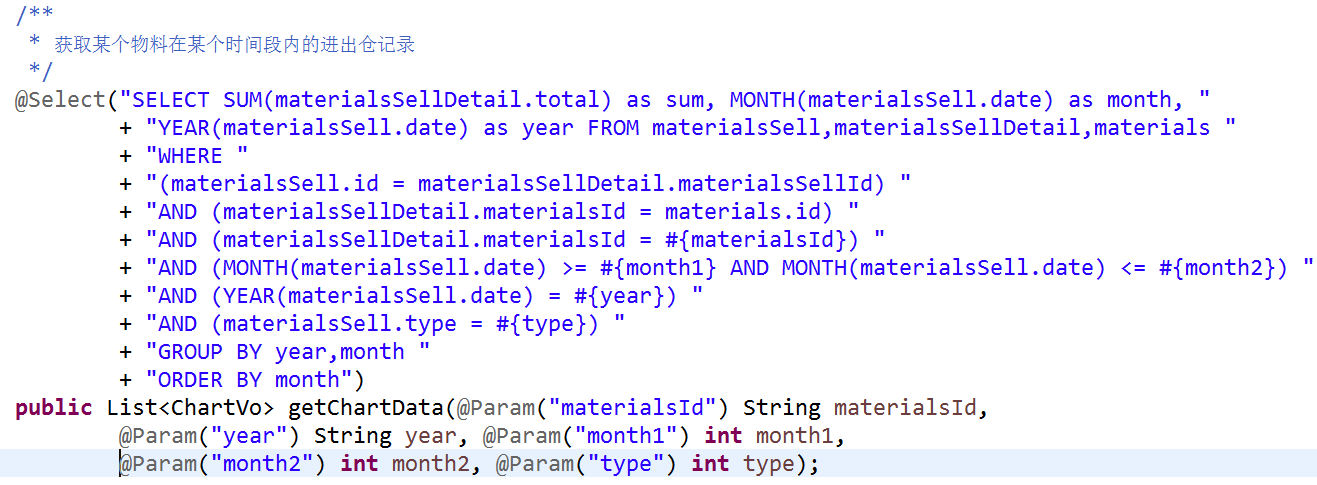
#### 1.2.3 主要流程



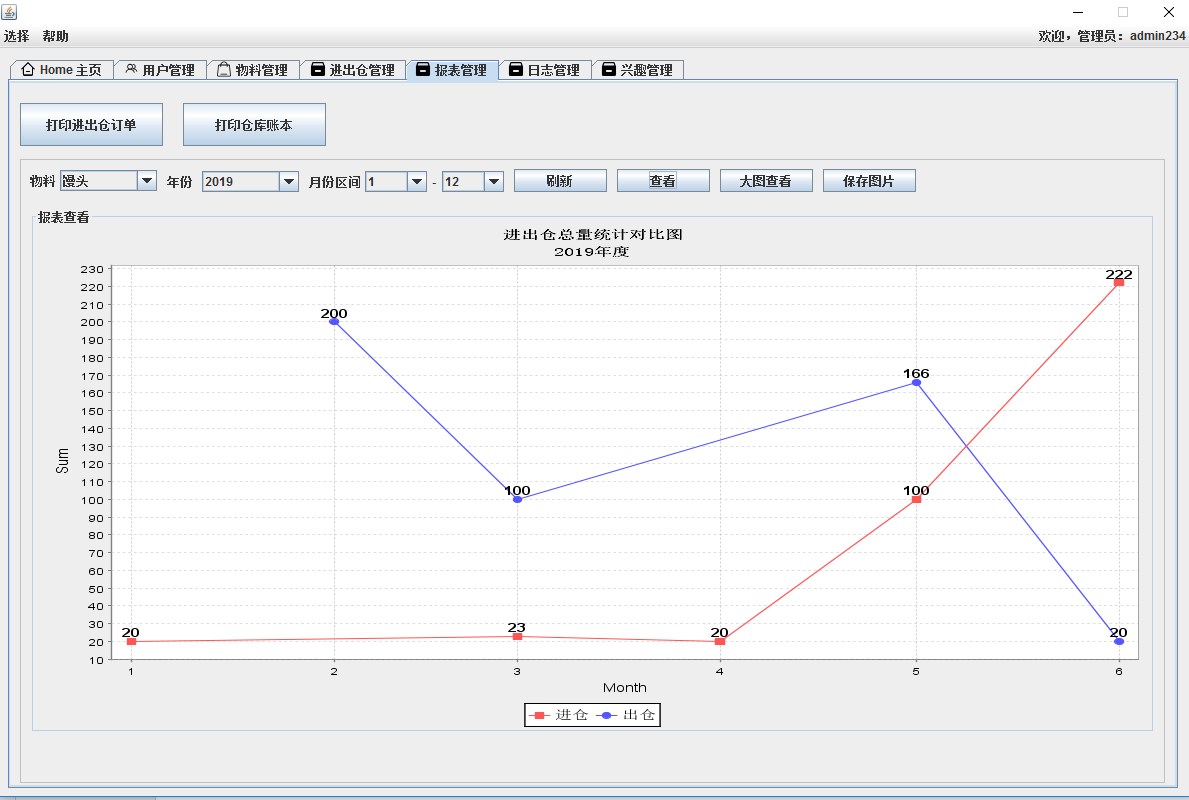
#### 1.2.4 主要数据库操作





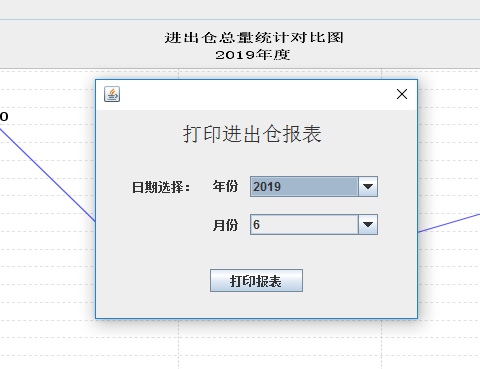


#### 1.2.5 主要测试



## 1.3 打印某个时间段所有进出仓订单信息

#### 1.3.1 具体界面



#### 1.3.2 实现思路

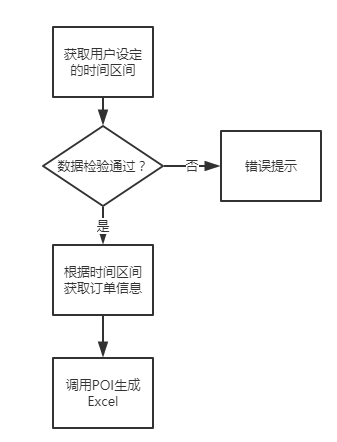
1、获取用户选择相应的时间区间

2、查询数据库获取相应的进出仓订单信息

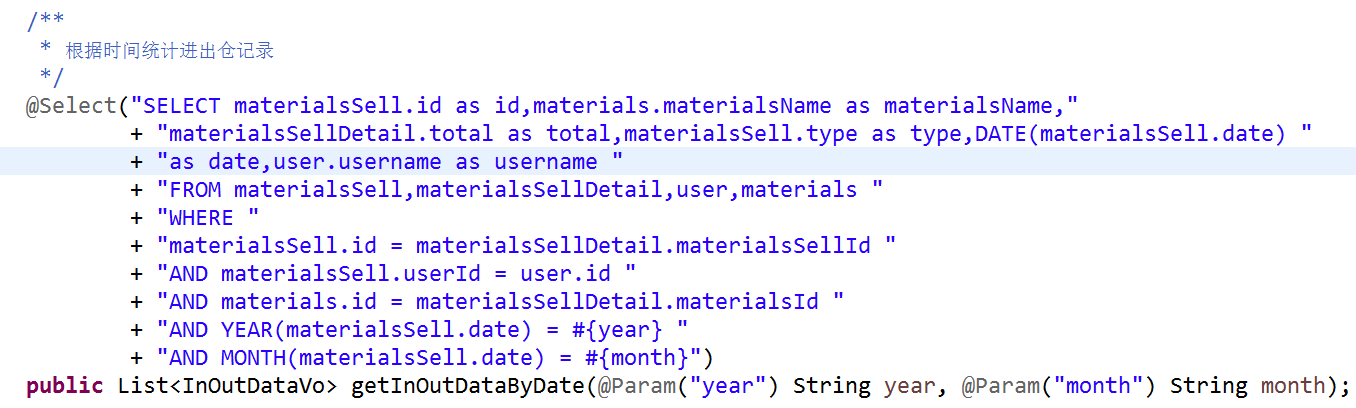
3、将查询的数据封装成具体的数据格式

4、调用POI程序打印Excel表格

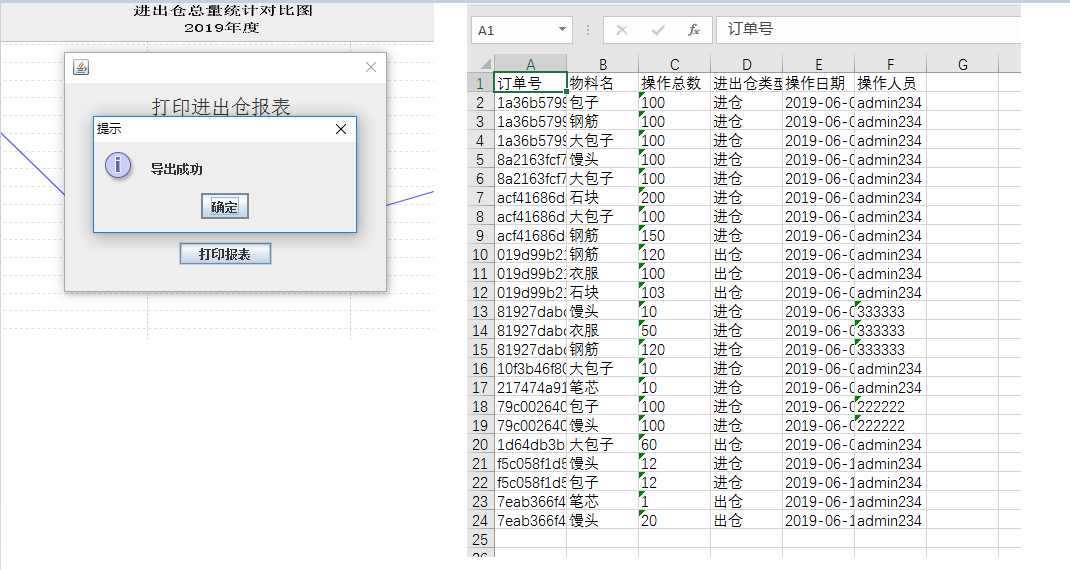
#### 1.3.3 具体流程



#### 1.3.4 主要数据库操作

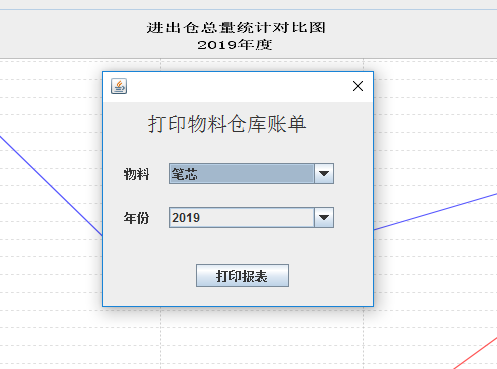


#### 1.3.5 主要测试



## 1.4 打印某个物料某个时间段内进出仓订单信息

#### 1.4.1 具体界面



#### 1.4.2 实现思路

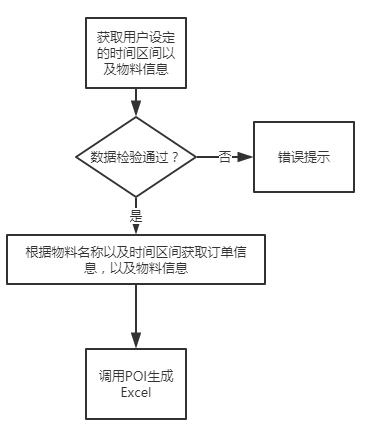
1、获取用户选择物料以及相应的时间区间

2、查询数据库获取相应的进出仓订单信息以及物料信息

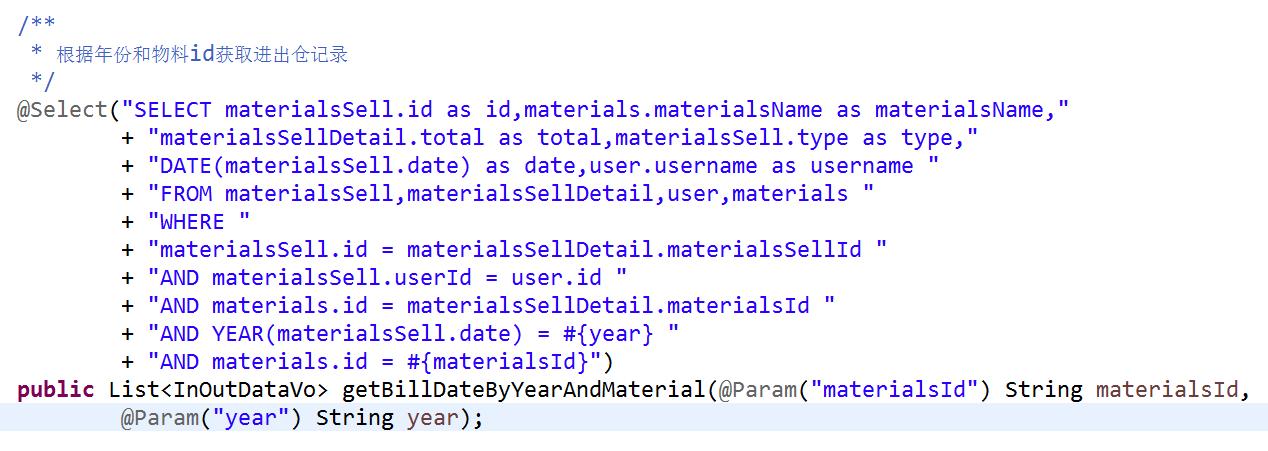
3、将查询的数据封装成具体的数据格式

4、调用POI程序打印Excel表格

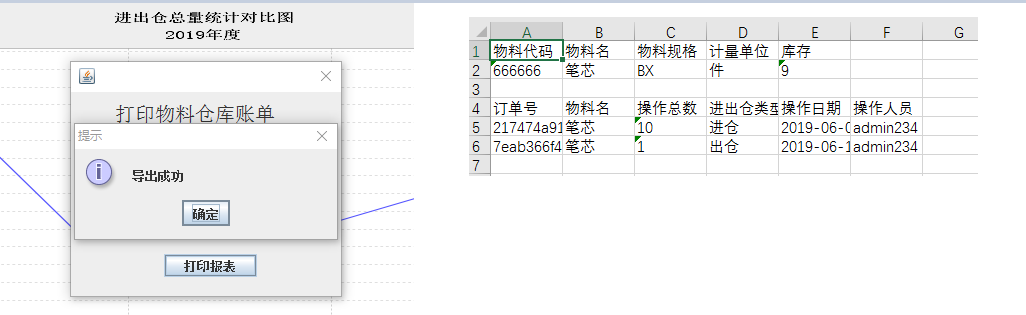
#### 1.4.3 主要流程



#### 1.4.4 主要数据库操作



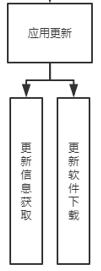
#### 1.4.5 主要测试



# 五、实验课题六：应用更新

## 1.0 主要需求

为了满足应用后期扩展的想法需要一个应用更新功能。



## 1.1 实现一个文件上传服务器用于保存更新文件以及更新信息

#### 1.1.1具体界面



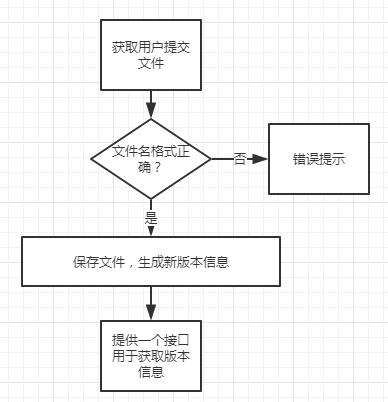
#### 1.1.2 实现思路

1、创建一个服务器获取上传文件，并保存

2、对上传文件规定上传命名格式用于获取版本信息 【文件上传名为xxxxx-版本.jar】

3、提供一个接口用于获取最新版本信息，包括版本号，文件名，文件大小，映射路径

#### 1.1.3 具体流程



#### 1.1.4主要测试

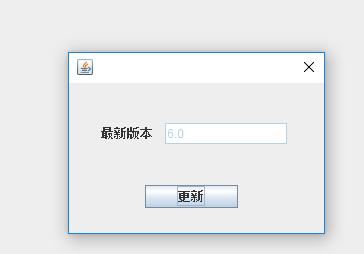


## 1.2 客户端与更新服务器端交互获取当前最新版本信息

#### 1.2.1具体界面







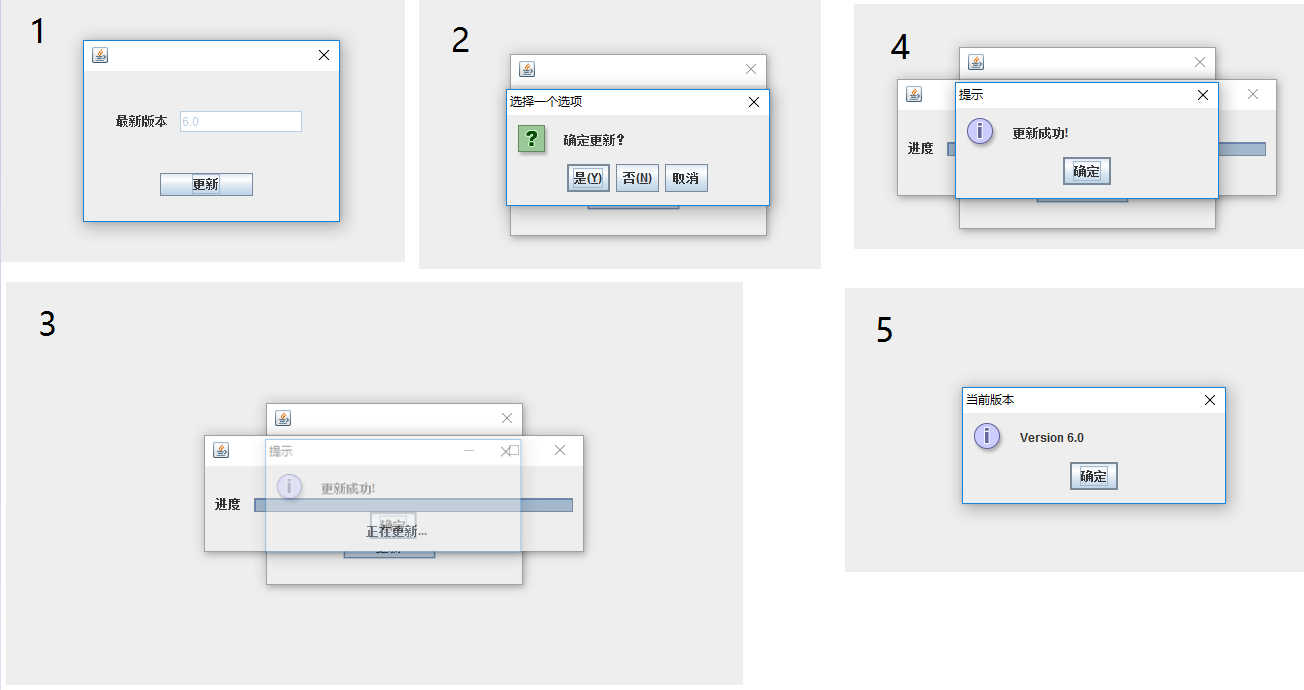
#### 1.2.2 实现思路

1、通过网络操作获取更新服务端接口信息

2、与当前版本比较，显示最新版本

## 1.3 实现一个更新程序，用于下载最新客户端

#### 1.3.1具体界面



#### 1.3.2 实现思路

1、与更新服务器端接口建立连接获取更新版本信息

2、提取信息中文件映射路劲信息

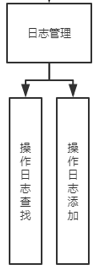
3、使用IO流下载新版本文件，并重命名

4、退出当前系统

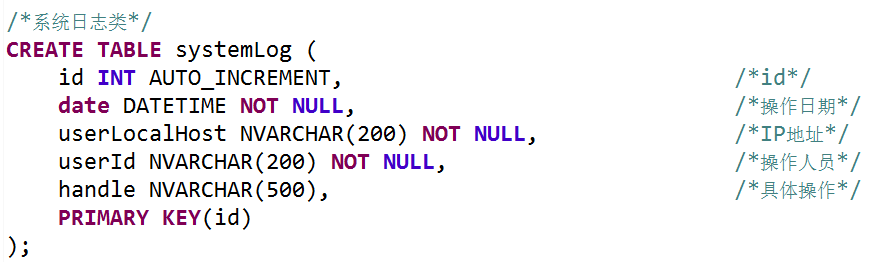
# 六、其他功能：日志管理

## 1.0 主要需求

为了防止员工的恶意操作以及方便管理员监察员工操作，使用开发出一套日志功能用来记录每个员工的所有操作



## 1.1 数据库表设计



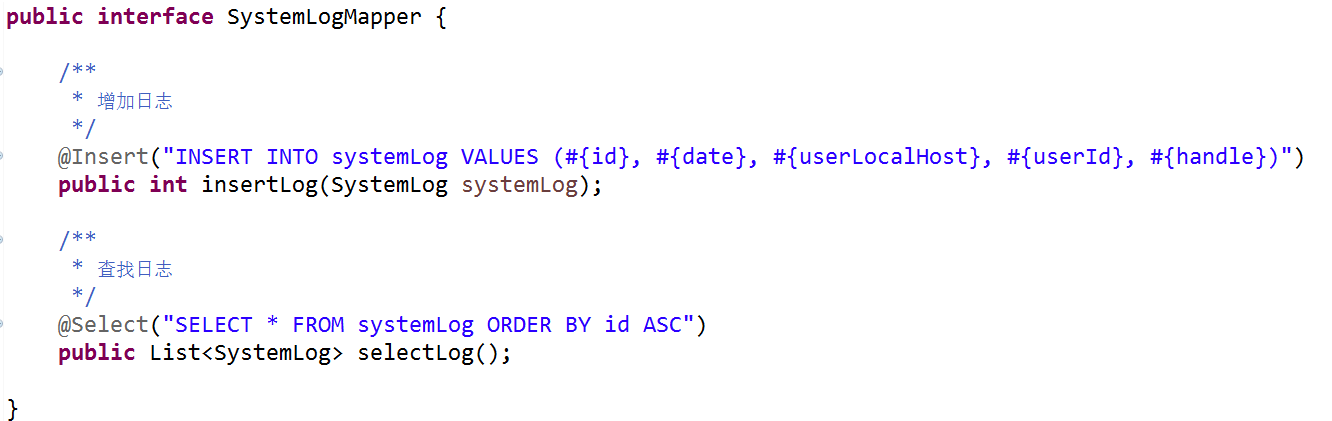
## 1.2 利用切面编程添加日志

#### 1.2.1 实现思路

1、使用Spring AOP技术在用户进行具体操作时获取当前登录用户信息，以及具体操作

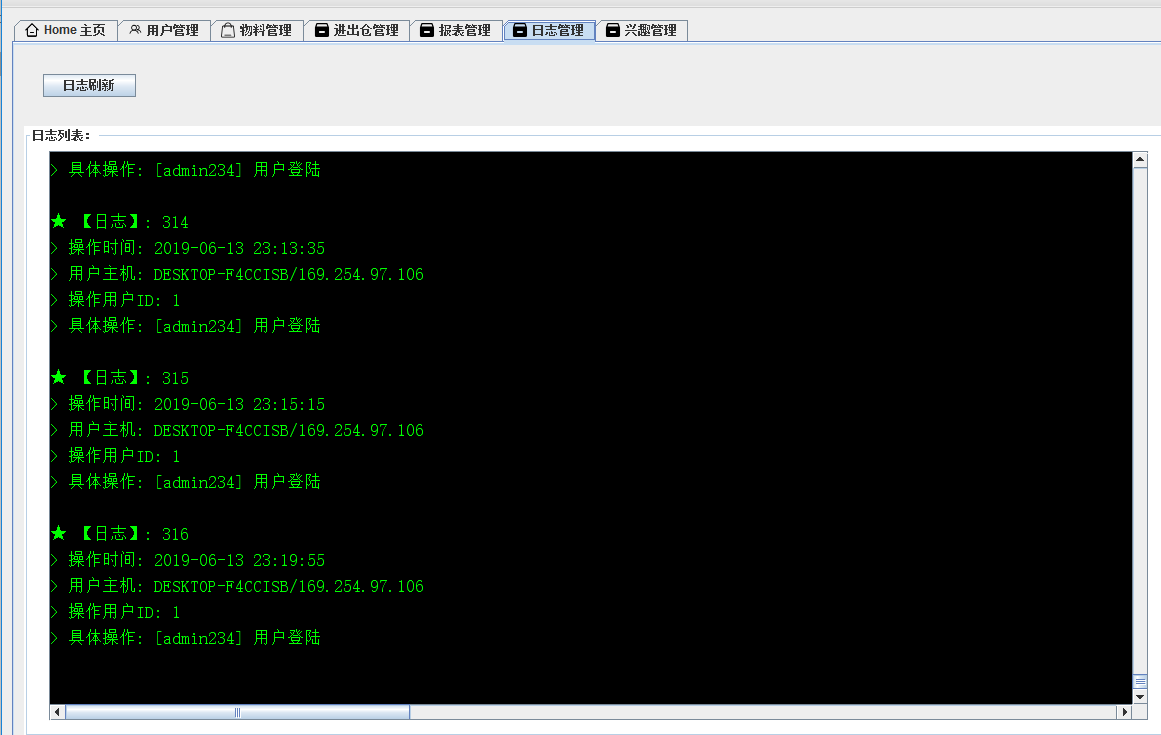
2、将获取到的信息进行格式化，保存到数据库

#### 1.2.3 主要数据库操作



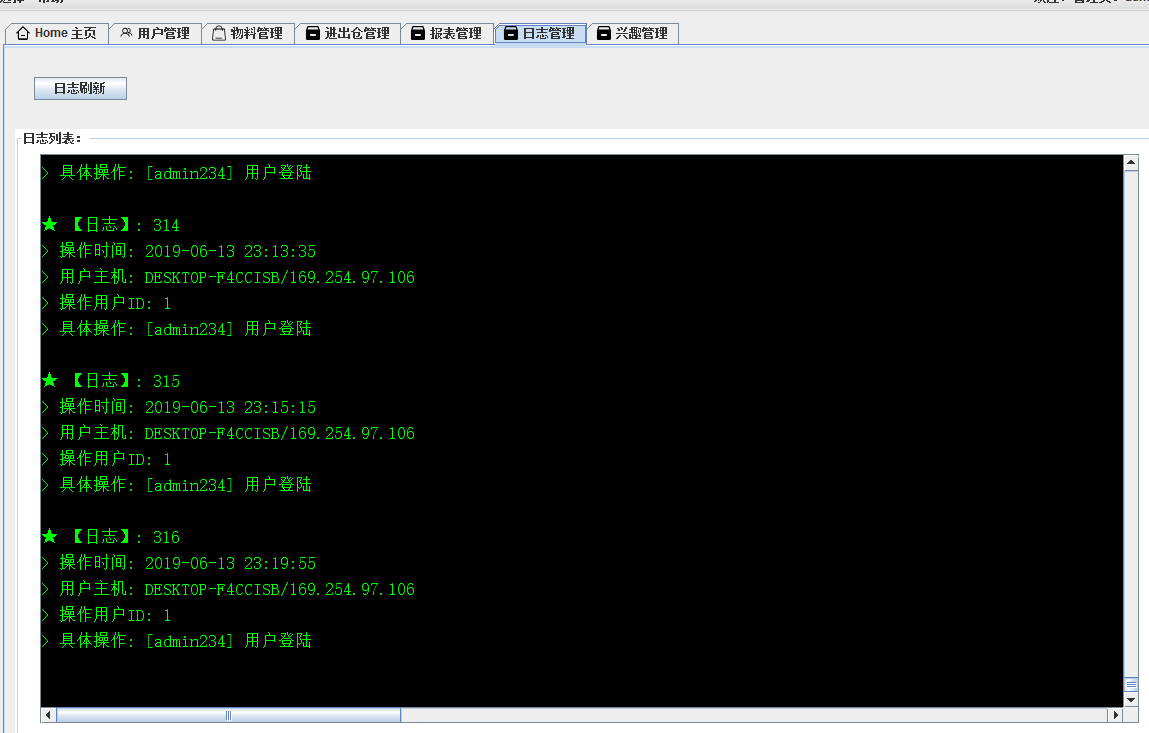
#### 1.2.4主要测试

当用户登陆时，日志记录登录信息：



## 1.3 日志查询

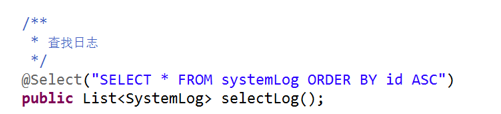
#### 1.3.1具体界面



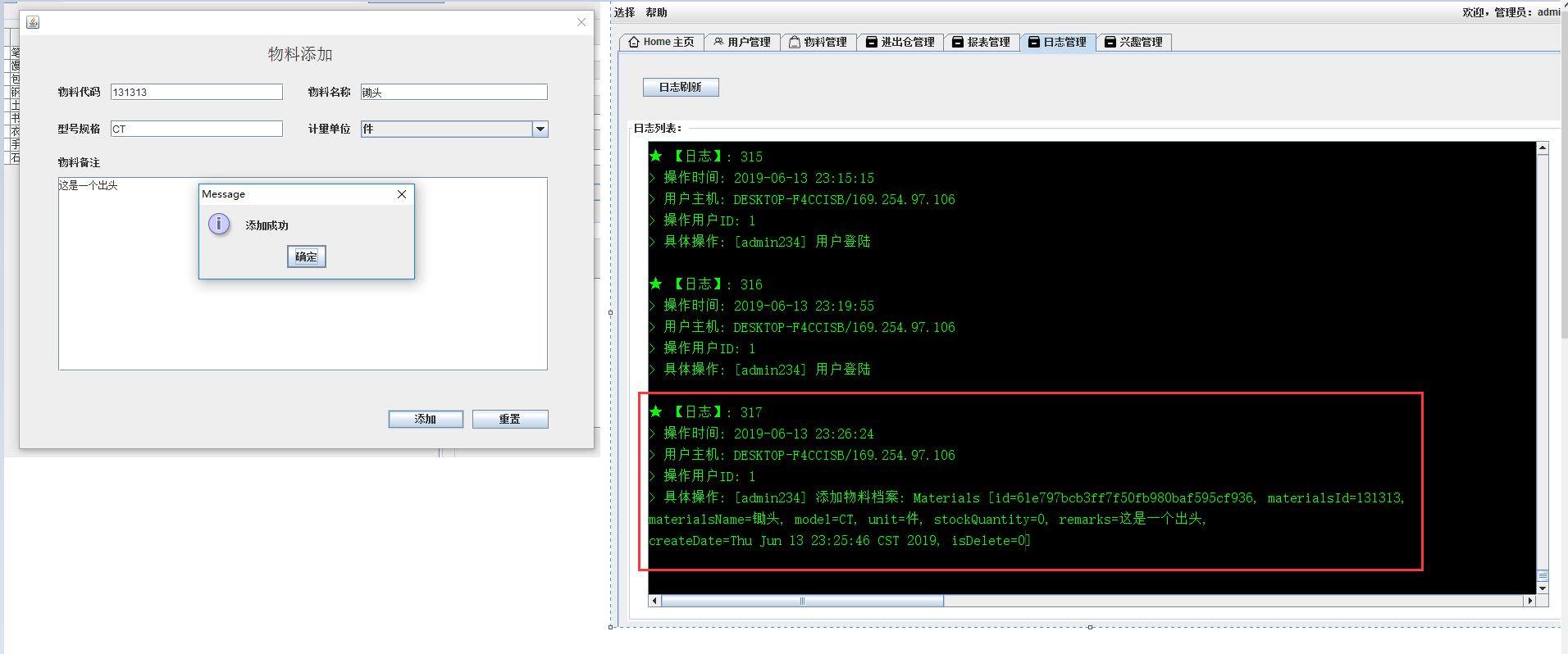
#### 1.3.2 实现思路

1、查询所有日志信息，按照时间排序

#### 1.3.3 主要数据库操作



#### 1.3.5主要测试



# [七、总结](#_Toc22993)

本次实验总体来说不是很难，本次实验是一个综合性很强的实验，不仅是对于技术的考察，更重要的是考察开发者能否考虑全面，本次实验中感觉有很多点没有做好：

1. 用户权限方面。未曾考虑到扩展性问题，对于该功能后续可以改进一下。
2. 查询功能以及记录展示方式：未作分页功能，以及自定义排序。
3. 应用更新：可以做的更加完善，比如可以添加历史纪录下载等

# 八、团队工作质量自评表

**应用软件开发》团队工作质量自评表**

团队工作质量分为总体质量和课题特定质量要求。

表1 总体开发质量表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 评价指标点 | 自评得分  （0-10） | 实现方式说明 | 教师  评分 |
| 1 | 窗体尺寸是否合适？包括窗体大小，长宽比等 | 9 | 固定窗口大小 |  |
| 2 | 窗体上控件布局是否整齐 | 9 | 使用绝对定位布局 |  |
| 3 | 可输入或可点击的控件是否设置了合理的TAB顺序 | 9 | 可以 |  |
| 4 | 界面功能实现是否符合用户的一般习惯（最好是不用开发者介绍或看手册就会使用） | 9 | 符合 |  |
| 5 | 变量名是否使用了与含义相关的命名？ | 9 | 是 |  |
| 6 | 是否有恰当的注释（方法名，参数，返回值，方法执行逻辑的解释，关键代码处） | 8 | 关键位置有 |  |
| 7 | 代码可读性（缩进对齐，代码易于理解） | 10 | 每行代码均有对齐 |  |

表2 课题特定的质量指标

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 评价指标点 | 自评得分  （0-10） | 实现方式说明 | 教师  评分 |
| 1 | 同一功能多个用户同时操作时是否会发生问题？（特别是多物料进出仓时） | 9 | 经过测试不会出现问题 |  |
| 2 | 是否实现了事务处理 | 9 | 是 |  |
| 3 | 功能实现方式是否考虑了表内数据很大时的性能表现 | 6 | 未考虑 |  |
| 4 | 性别用“男”、“女”选择方式 | 10 | 单选按钮 |  |
| 5 | 出生日期必须用日期控件来控制 | 10 | 日期控件来控制 |  |
| 6 | 修改、删除程序在执行修改删除操作前，都必须进行记录定位操作 | 8 | 大都有进行相应的查询 |  |
| 7 | 使用模糊查询 | 9 | 有 |  |
| 8 | 进出仓单号自动生成 | 9 | 采用32位随机值的自动生成方式 |  |
| 9 | 使用存储过程 | 0 | 无存储过程 |  |
| 10 | 权限管理系统：是否可以增加或减少管理的权限总数 | 10 | 可以 |  |
| 11 | 是否可以对用户进行授权或回收权限 | 10 | 可以回收与授权 |  |
| 12 | 是否实现了用户权限组 | 9 | 是 |  |
| 13 | 报表：是否可打印任意月份的报表 | 9 | 可以 |  |
| 14 | 是否有图形状报表 | 9 | 折线图报表 |  |
| 15 | 是否有打印出来的纸质或电子报表的样本 | 9 | 可以打印成EXCEL表格，存放在本地目录中 |  |
| 16 | 打印布局是否合理 | 9 | 合理 |  |

文档质量指标

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 评价指标点 | 自评得分  （0-10） | 实现情况说明 | 教师  评分 |
| 1 | 页数 | 9 | 65 |  |
| 2 | 流程图 | 9 | 都有具体流程图 |  |
| 3 | 设计思想描述 | 8 | 都有具体的实现思路描述 |  |
| 4 | 典型代码 | 9 | 都有附上具体的数据库操作代码 |  |
| 5 | 实验过程记录 | 8 | 有 |  |
| 6 | 心得体会 | 8 | 有 |  |

# 九、团队工作量自评表

团队编号： 1 成员：林国祥、张波、于鉴琪

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 工作项目 | 工作量(代码行数) | 林国祥 | 张波 | 于鉴琪 |
| SystemLogMaterialsAspect.java | 1 | 编写者 |  |  |
| SystemLogUserAspect.java | 1 | 编写者 |  |  |
| SystemLogWarehouseAspect.java | 1 | 编写者 |  |  |
| MaterialsConst.java | 3 | 编写者 |  |  |
| UserConst.java | 3 | 编写者 |  |  |
| WarehouseConst.java | 1 | 编写者 |  |  |
| MaterialsAspectHandler.java | 20 | 编写者 |  |  |
| UserAspectHandler.java | 30 | 编写者 |  |  |
| SystemLogMapper.java | 4 | 编写者 |  |  |
| ChartVo.java | 36 | 编写者 |  |  |
| ClientMSG.java | 4 | 编写者 |  |  |
| ComboxVo.java | 15 | 编写者 |  |  |
| Session.java | 5 | 编写者 |  |  |
| SystemLog.java | 40 | 编写者 |  |  |
| Version.java | 10 | 编写者 |  |  |
| SystemLogServiceImpl.java | 100 | 编写者 |  |  |
| SystemLogService.java | 16 | 编写者 |  |  |
| MD5Util.java | 30 | 编写者 |  |  |
| POIUtil.java | 40 | 编写者 |  |  |
| SpringContextUtils.java | 30 | 编写者 |  |  |
| SystemUtil.java | 20 | 编写者 |  |  |
| UpdateUtil.java | 20 | 编写者 |  |  |
| ChartComponent.java | 100 | 编写者 |  |  |
| FormManagerPanel.java | 190 | 编写者 |  |  |
| LogManagerPanel.java | 40 | 编写者 |  |  |
| MaterialManagerPanel.java | 300 | 编写者 |  |  |
| TableComponent.java | 30 | 编写者 |  |  |
| MaterialsBillPrintPage.java | 100 | 编写者 |  |  |
| MaterialsInOutPrintPage.java | 100 | 编写者 |  |  |
| UpdatePage.java | 50 | 编写者 |  |  |
| ViewStart.java | 2 | 编写者 |  |  |
| WgClientSimpleApplication.java | 4 | 编写者 |  |  |
| MultipartConfig.java | 2 | 编写者 |  |  |
| WebMvcConfig.java | 2 | 编写者 |  |  |
| FileHandlerController.java | 50 | 编写者 |  |  |
| PageHandlerController.java | 4 | 编写者 |  |  |
| ResultInfo.java | 10 | 编写者 |  |  |
| Version.java | 10 | 编写者 |  |  |
| MD5Util.java | 10 | 编写者 |  |  |
| WgServerSimpleApplication.java | 4 | 编写者 |  |  |
| Update.java | 10 | 编写者 |  |  |
| Version.java | 10 | 编写者 |  |  |
| WgUpdateSimpleApplication.java | 4 | 编写者 |  |  |
| MaterialsService.java | 16 | 编写者 |  |  |
| UserService.java | 16 | 编写者 |  |  |
| MaterialsSellDetailService.java | 2 | 编写者 |  |  |
| MaterialsSellService.java | 16 | 编写者 |  |  |
| MaterialsSellDetailMapper.java | 12 | 可答辩 | 可答辩 | 编写者 |
| MaterialsSellMapper.java | 20 | 可答辩 | 可答辩 | 编写者 |
| MaterialsSellDetailVo.java | 10 | 可答辩 | 可答辩 | 编写者 |
| InOutDataVo.java | 40 | 可答辩 |  | 编写者 |
| MaterialsSellVo.java | 20 | 可答辩 |  | 编写者 |
| MaterialsSell.java | 40 | 可答辩 | 可答辩 | 编写者 |
| MaterialsSellDetail.java | 40 | 可答辩 | 可答辩 | 编写者 |
| MaterialInDetailPanel.java | 90 | 可答辩 | 可答辩 | 编写者 |
| MaterialOutDetailPanel.java | 120 | 可答辩 | 可答辩 | 编写者 |
| TradingManagerPanel.java | 250 | 可答辩 | 可答辩 | 编写者 |
| MaterialInPage.java | 200 | 可答辩 | 可答辩 | 编写者 |
| MaterialOutPage.java | 170 | 可答辩 | 可答辩 | 编写者 |
| RemarksPage.java | 40 | 可答辩 |  | 编写者 |
| IndexPanel.java | 10 | 可答辩 |  | 编写者 |
| InterestManagerPanel.java | 10 | 可答辩 | 可答辩 | 编写者 |
| MaterialsSellDetailServiceImpl.java | 15 | 可答辩 | 可答辩 | 编写者 |
| MaterialsSellServiceImpl.java | 150 | 可答辩 | 可答辩 | 编写者 |
| Materials.java | 4 | 可答辩 | 可答辩 | 编写者 |
| DataUtil.java | 30 | 可答辩 |  | 编写者 |
| db.sql | 60 | 可答辩 |  | 编写者 |
| DateUtil.java | 30 | 可答辩 | 可答辩 | 编写者 |
| DateComponent.java | 90 | 可答辩 | 可答辩 | 编写者 |
| MaterialAddPage.java | 120 | 可答辩 | 可答辩 | 编写者 |
| User.java | 40 | 可答辩 | 编写者 | 可答辩 |
| HomePage.java | 130 | 可答辩 | 编写者 | 可答辩 |
| LoginPage.java | 100 | 可答辩 | 编写者 | 可答辩 |
| MaterialAddPage.java | 120 | 可答辩 | 编写者 | 可答辩 |
| UserAddPage.java | 300 | 可答辩 | 编写者 | 可答辩 |
| UserUpdatePage.java | 220 | 可答辩 | 编写者 | 可答辩 |
| MaterialsServiceImpl.java | 100 | 可答辩 | 编写者 | 可答辩 |
| UserServiceImpl.java | 100 | 可答辩 | 编写者 | 可答辩 |
| MaterialsMapper.java | 16 | 可答辩 | 编写者 | 可答辩 |
| UserMapper.java | 14 | 可答辩 | 编写者 | 可答辩 |
| UserManagerPanel.java | 500 | 可答辩 | 编写者 | 可答辩 |
|  |  |  |  |  |
| 文档1 |  |  |  |  |
| 小组代码总行数 | 4689 |  |  |  |
| 成员工作量  （文档+代码） | 编写代码总行数（不少于小组代码总行数的20%） | 1562 | 1520 | 1607 |
| 可答辩代码总行数 （本人 “可答辩”代码之和） | 4689 | 2891 | 3127 |
| 可用于答辩的代码量（前两项之和，不低于小组总代码行数的40%或不低于于1000行代码）及占小组总代码行数的比例 | 4689 | 2891 | 3127 |
| 折算代码量（本人编写的代码总行数+可答辩代码总行数\*0.3） | 1562+4689\*0.3=2968.7 | 1520+2891\*0.3=2387 | 1607+3127\*0.3=2545 |
| 文档页数 |  |  |  |

注：成员对于一个模块的贡献可分为“编写者”和“可答辩”，编写者为代码的实际编写者，可答辩表示虽然代码不是自己写，但参与了代码的编写，熟悉其内容，可以就该代码进行答辩。

可答辩

成员工作量计算：

代码量 ： 折算代码量=自己编写代码量+可答辩代码量\*0.3

# [十、实验的源程序代码](#_Toc684)

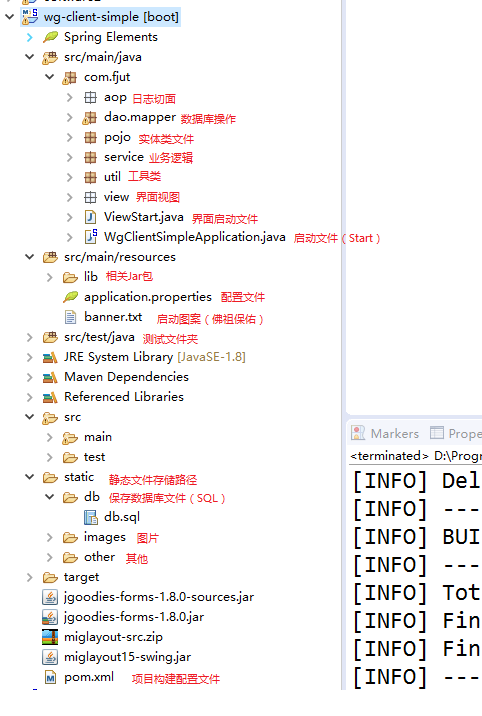
## 1、主要项目功能介绍

wg-client-simple 应用客户端

wg-server-simple 文件上传服务器

wg-update-simple 应用更新程序

## 2、工程文件结构



## 3、具体代码详见附带压缩包