**江南大学**

**物联网工程学院**

**课程设计报告**

课程： 软件工程

题目： 食堂智能管理系统

专业： 物联网工程

班级： 物联网1704

成员： 倪李剑

完成日期： 2019.12.5

**食堂智能管理系统**

【摘要】随着信息技术的飞速发展、互联网的普及，使得计算机、网络技术已经快速融入到校园的需求中。传统的食堂采购管理已经不能满足现今越来越追求高效速度的生活理念了，学校食堂需要一个更加智能化更加便利快捷的智能采购平台。为此，食堂智能采购平台的设计需求促动着本次平台的设计。

本文描述了一个江南大学的食堂智能采购平台的设计开发过程，依据面向对象设计的开发思想，运用纯JAVA代码和mySQL实现智能采购平台的设计。系统介绍了在JAVA和MySQL数据库的设计作用下，阐述了系统的设计方案、实现方法以及所采用的开发工具和相关技术。设计的食堂智能采购系统仅供食堂的工作人员使用。系统共有仓库出入库管理、采购订货、查询系统。

**关键词**：**纯Java代码**；MySQL；智能化；食堂智能采购平台

**目录**

[1.项目背景说明 2](#_Toc5277)

[2.开发技术 3](#_Toc15471)

[开发环境（纯java开发） 3](#_Toc17463)

[2.1 Java语言（本项目是由纯Java开发） 3](#_Toc13336)

[2.2 MySQL数据库 4](#_Toc32758)

[3.可行性分析 4](#_Toc30283)

[3.1 技术可行性分析 5](#_Toc16431)

[3.2经济可行性分析 5](#_Toc27593)

[3.3社会可行性分析 5](#_Toc23897)

[4.需求分析 5](#_Toc15107)

[4.1 业务需求分析 5](#_Toc7338)

[4.2用户需求分析 5](#_Toc4785)

[4.3功能需求分析 6](#_Toc32466)

[4.4非功能需求分析 6](#_Toc12862)

[5.系统概要设计 6](#_Toc18752)

[5.1系统功能模块设计 6](#_Toc26204)

[5.2系统流程图设计 7](#_Toc18128)

[5.3系统数据库设计 8](#_Toc9092)

[6.系统详细设计 10](#_Toc25320)

[6.1界面设计 10](#_Toc29999)

[（1）用户功能模块 10](#_Toc6709)

[（2）主界面 11](#_Toc28673)

[（3）各个功能模块 11](#_Toc22195)

[7.系统测试 18](#_Toc1611)

[（1） 登陆模块测试如下： 18](#_Toc14506)

[（2）出入库模块测试 18](#_Toc22918)

[（3）查询模块 19](#_Toc7354)

[（4）订购模块 19](#_Toc14517)

[8. 问题分析和总结 19](#_Toc28469)

**1.项目背景说明**

电子商务是以信息网络技术为手段，以商品交换为中心的商务活动，以电子交易方式进行交易活动和相关服务的活动，是传统商务各环节的电子化、网络化、信息化。

对于公司以及学校来说，食堂管理显得十分重要，随着我国经济的发展，我国的食堂管理方案也在不断迅速的发展，但也存在许多问题，如管理方法落后、食材查询不方便、资源整合率低等问题。随着物联网技术和计算机技术的不断发展，是越来越多的食物与计算机和网络息息相关，很多工作可使用更加高效的信息化方式进行。

传统的食堂管理方式在物联网时代可以以更加高效率的方式进行。智能食堂管理系统为传统的食堂提供一种新的经营方式。其主要的优点有如下：

1. 食堂可以方便的管理食堂食材余量。
2. 界面简洁，易于上手；
3. 基于数据库系统，便于数据更新管理；
4. 所需管理人员较少，人力成本低；
5. 可根据不同场景更改数据库信息，适应不同条件、环境，更为灵活；

**1.1 项目问题描述**

本项目主要是应用物联网的基本原理，使用java搭建一个智能食堂管理系统。该系统主要包括用户登录，食材的入库出库登记，录入食材，查询所有食材，查询单个食材数量，查询所有出入库信息，查看商家信息，在线订购食材等功能。

**1.2 目的及意义**

随着物联网的发展，人们的生活方式及需求也在发证变化，食堂管理更加依赖于机器的记录，而不是人工的录入，食材的订购也更加以来于网络上的商家。 根据而智能食堂管理系统可以给食堂减轻负担，满足食堂对智能管理和智能订购的需求，节省了食堂人工的费用以及管理的时间。

**2.开发技术**

开发环境（纯java开发）

Windows

Eclipse（java）

Navicat for mysql（mysql）

**2.1 Java语言（本项目是由纯Java开发）**

java是一门面向对象编程语言，不仅吸收了C++语言的各种优点，还摒弃了C++里难以理解的多继承、指针等概念，因此Java语言具有功能强大和简单易用两个特征。Java语言作为静态面向对象编程语言的代表，极好地实现了面向对象理论，允许程序员以优雅的思维方式进行复杂的编程。Java具有简单性、面向对象、分布式、健壮性、安全性、平台独立与可移植性、多线程、动态性等特点。Java可以编写桌面应用程序、Web应用程序、分布式系统和嵌入式系统应用程序等。

（1）Java语言提供了对内存的自动管理，程序员无需在程序中进行分配、释放内存，那些可怕的内存分配错误不会再打扰你了；

（2）去除了C++语言中的令人费解、容易出错的“指针”，用其它方法来进行弥补；

（3）避免了赋值语句（如a = 3）与逻辑运算语句（如a = = 3）的混淆；

（4）取消了多重继承这一复杂的设施。

java语言的优点：

1. 简洁有效；2.高可移植性；3.“面向对象”；4.解释型；5.适合分布式计算；6.拥有较好的性能；7.健壮、防患于未然；8.具有多线程处理能力；9.具有较高的安全性；10.是一种动态语言；11.中性的对象语言。

同时配合java使用的是Eclipse。Eclipse是一个开放源代码的、基于Java的可扩展开发平台。

**2.2 MySQL数据库**

MySQL是一种开放源代码的关系型数据库管理系统（RDBMS），使用最常用的数据库管理语言--结构化查询语言（SQL）进行数据库管理。MySQL是开放源代码的，因此任何人都可以在General Public License的许可下下载并根据个性化的需要对其进行修改。MySQL因为其速度、可靠性和适应性而备受关注。大多数人都认为在不需要事务化处理的情况下，MySQL是管理内容最好的选择。

MySQL数据库的特点：

（1）MySQL数据库是用C和C++语言编写的，并且使用了多种编辑器进行测试，以保证源码的可移植性。

（2）支持多个操作系统例如：Windows、Linux、Mac OS等等。

（3）支持多线程，可以充分的利用CPU资源。

（4）为多种编程语言提供API，包括C语言，Java，PHP。Python语言等。

（5）MySQL优化了SQL算法，有效的提高了查询速度。

（6）MySQL内提供了用于管理，检查以及优化数据库操作的管理工具。

（7）它能够作为一个单独的应用程序应用在客户端服务器网络环境中，也可以作为一个库嵌入到其他的软件中并提供多种语言支持。

MySQL数据库能做什么：

（1）它可以存储大量的数据，方便用户检索和访问。

（2）用于保存信息数据以及信息的一致性和完整性。

（3）数据的共享和安全。

（4）通过组合分析，产生有用信息。

**3.可行性分析**

结合目前的实际情况，分析一个系统开发的必要性和合理性，即使该系统能够进行项目开发的可行性，需要从经济、技术和社会可行性等条件进行分析。

**3.1 技术可行性分析**

本智能管理系统考虑到硬件和软件环境的要求，采用的是Eclipse开发平台和Navicat for mysql（mysql）。使用Eclipse作为开发平台，能够帮助开发者提高来发软件的效率和质量，极大程度的缩短开发周期。Navicat for mysql是MySQL的可视化工具，可以帮助开发者更加方便的建立数据库。本系统使用的是纯Java开发。

**3.2经济可行性分析**

本智能食堂管理系统使用的所有软件开发平台，都是免费使用的，也可以从网上搜索得到，然后惊醒环境配置和平台搭建，就可以使用。从经济上，本设计没有支付任何开发平台的费用，即使后期与商店的合作也只需要很伤的经济投入。

**3.3社会可行性分析**

智能食堂管理系统是物联网的部分，紧跟社会发展的趋势，是将来食堂管理的必需，必然被社会接受。

综上，智能食堂管理系统的开发具有可行性。

**4.需求分析**

一个软件在开发之前，需求分析是非常关键的步骤，该过程一般分为功能分析和非功能需求分析。

**4.1 业务需求分析**

传统食堂管理只能依靠人工来计算，而另一方面食材购买比较麻烦，因而本系统主要实现食堂管理的智能化。食堂管理员可以直接进行食材的入库出库、可以进行查询食材余量、直接在网上进行食材的订购等等功能。

**4.2用户需求分析**

本智能食堂管理系统主要是满足的人群是食堂管理员以及食材提供商。对于食堂管理者来说，可以帮助他们减小食堂管理的人力资本，更加方便的管理食堂仓库。

对于食材供应商来说，他们能够更加直接的收到食材的订单，更加方便的能够分配食材。

**4.3功能需求分析**

本系统主要是为食堂管理员减轻负担，对食堂的食材进行管理查询操作。

通过功能需求分析，对智能食堂管理系统的前期需求进行了如下定义：

1. 用户的登陆功能，识别用户功能。
2. 用户可以进行食材的入库出库操作
3. 用户可以对仓库所有食材进行查询，统计所有的余量。
4. 用户可以对出入库的所有信息进行查询，方便用户在出错的时候进行查询来寻找出错的订单。
5. 用户可以在网上直接进行采购食材。
6. 用户可以对采购信息进行查询。

**4.4非功能需求分析**

1. 安全可靠行需求。系统安全性和可靠性是必须要考虑的。
2. 数据处理能力。高效的数据处理能够提高系统的相应速度。

（3）易维护性和实用性。一个系统开发要满足用户的需求并且要具有良好的维护性。

**5.系统概要设计**

## 5.1系统功能模块设计

根据前期的需求分析，主要分为用户管理模块、出入库管理模块、采购模块和查询模块。其中用户模块主要包括用户登陆以及用户管理。出入库模块主要包括出入库信息和食物录入。采购模块主要包括下单模块和选择商户。查询模块想主要包括查询单一食材、查询所有食材、查询出入库信息和查询采购信息。

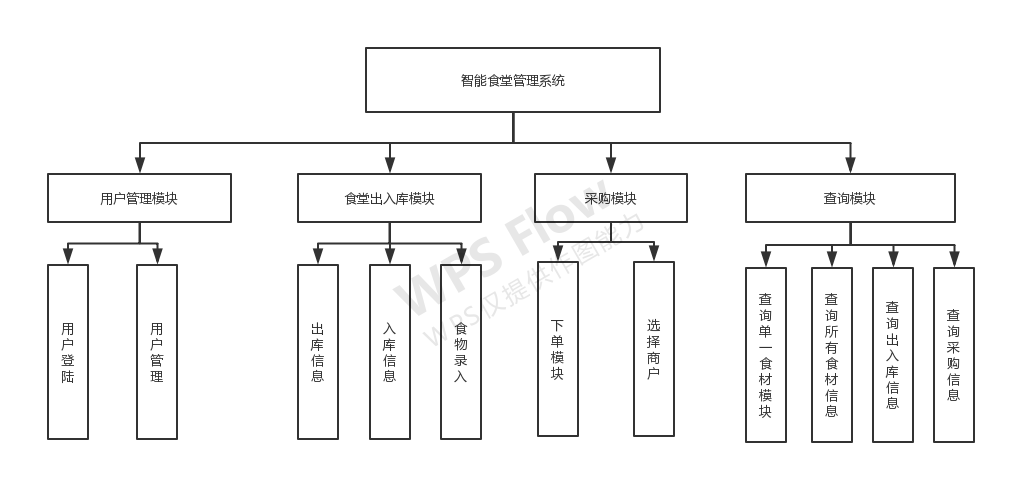
****

图1 系统功能模块图

## 5.2系统流程图设计

本系统主要业务流程为：用户必须先通过验证才能登陆系统，进入仓库管理，进行出入库管理；在采购界面进行食材的订购；在查询界面进行各种查询操作。

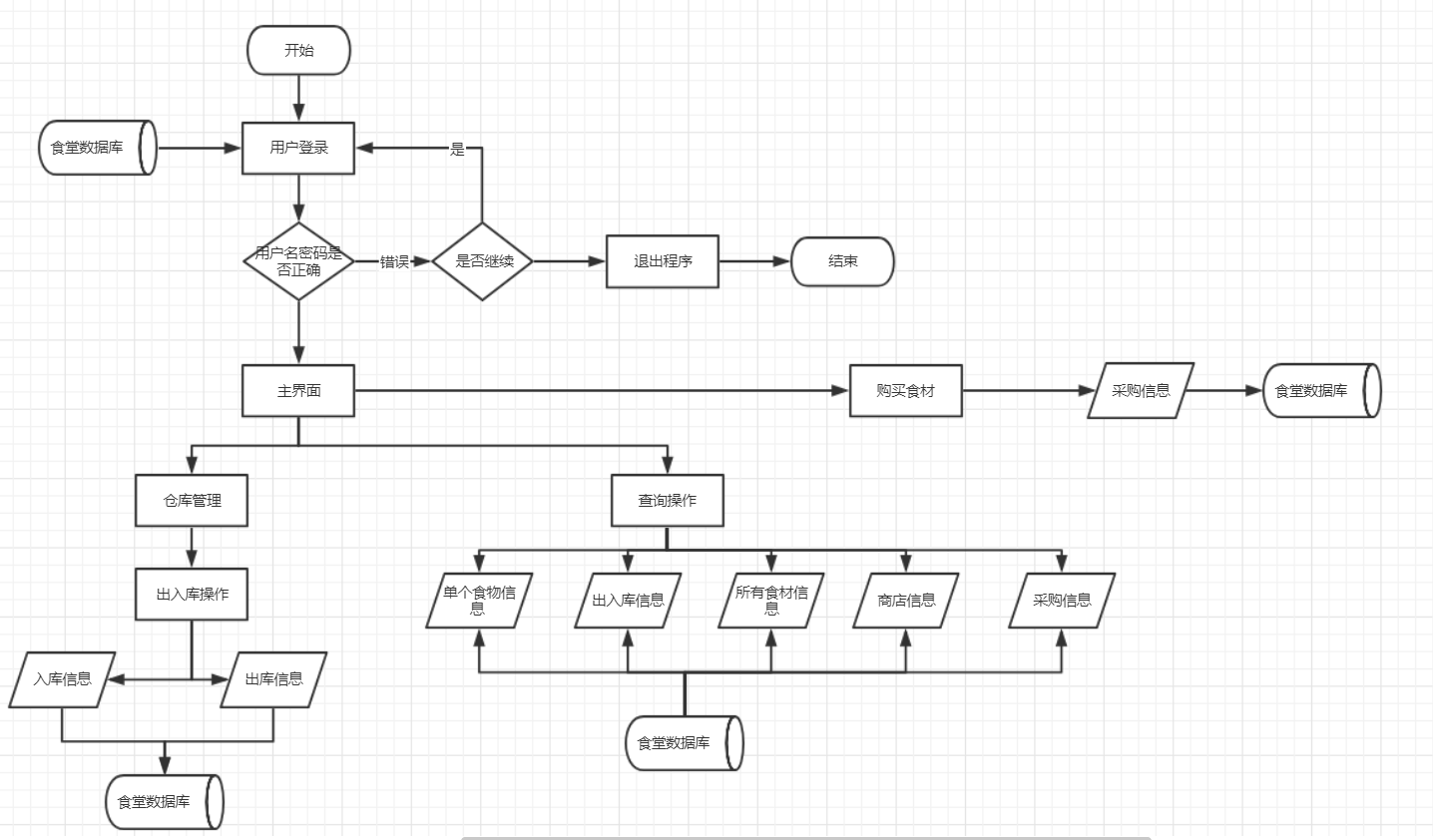


图2 系统流程图

## 5.3系统数据库设计

数据是一个管理系统的核心，只有一个好的数据储存结构才能保证数据操作的高效性和快速性，因此根据系统的需求绘制了系统的E-R图，如图3所示。

针对前面的需求分析，设置了用户信息表、食物信息表、商店信息表、出入库信息表、采购信息表。

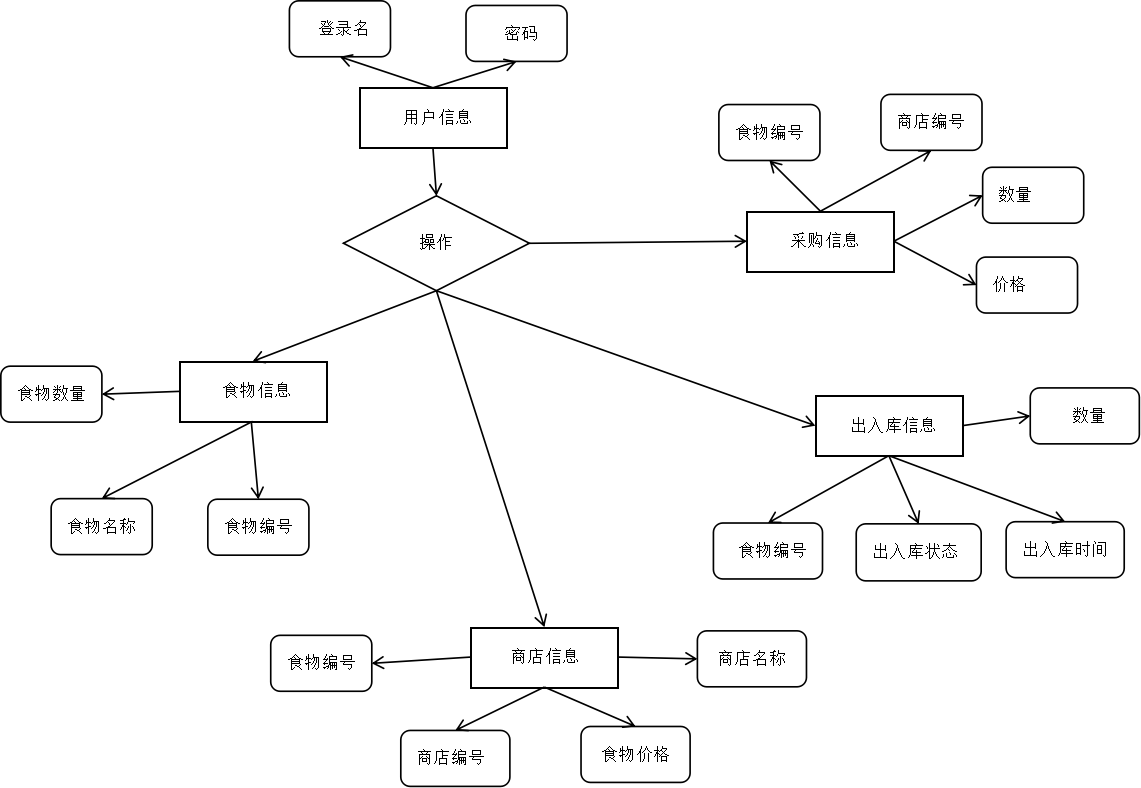
****

图3 系统E-R图

数据字典如下所示

表**inorout\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **类型** | **长度** | **允许空值** | **小数点** | **备注** |
| Food\_number | char | 10 | 否 | 0 | 食物编号 |
| status | Char | 6 | 否 | 0 | 出入库状态 |
| Amount | Int | 16 | 否 | 0 | 数量 |
| time | Datetime | 0 | 否 | 0 | 出入库时间 |

表**food\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **类型** | **长度** | **允许空值** | **小数点** | **备注** |
| Food\_number | Char | 10 | 否 | 0 | 食物编号 |
| Food\_name | Char | 16 | 否 | 0 | 食物名称 |
| Food\_amount | int | 12 | 否 | 0 | 食物数量 |

表**login**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **类型** | **长度** | **允许空值** | **小数点** | **备注** |
| Id | Char | 5 | 否 | 0 | 登陆名 |
| Password | Char | 6 | 否 | 0 | 登录密码 |

表**store\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **类型** | **长度** | **允许空值** | **小数点** | **备注** |
| store\_number | Char | 12 | 否 | 0 | 商店编号 |
| store\_name | Char | 30 | 否 | 0 | 商店名称 |

表**purchase**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **类型** | **长度** | **允许空值** | **小数点** | **备注** |
| store\_number | Char | 12 | 否 | 0 | 商店编号 |
| Food\_number | Char | 16 | 否 | 0 | 食物编号 |
| amount | int | 12 | 否 | 0 | 食物数量 |
| Food\_name | Char | 16 | 否 | 0 | 食物名称 |
| Price | Float | 12 | 否 | 0 | 价格 |
| Time | date time |  |  |  | 购买时间 |

表**food\_info**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **字段名** | **类型** | **长度** | **允许空值** | **小数点** | **备注** |
| Food\_number | Char | 10 | 否 | 0 | 食物编号 |
| Store\_number | Char | 16 | 否 | 0 | 商店编号 |
| Price | Float | 10 | 否 | 0 | 食物价格 |

**6.系统详细设计**

本系统需要与数据库进行连接，使用JDBC实现java和mysql数据库的连接。

driver=com.mysql.cj.jdbc.Driver

url=jdbc:mysql://localhost:3306/food ? characterEncoding=utf8 & useSSL=false & serverTimezone=Asia/Shanghai & rewriteBatchedStatements=true

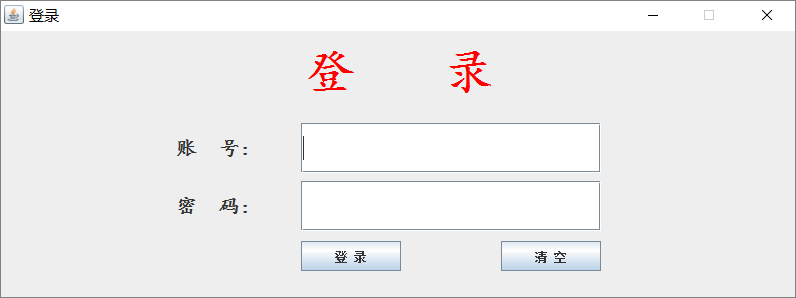
user=root

password=

**6.1界面设计**

**（1）用户功能模块**

登陆界面如下：



若用户名过着密码错误，会提示登录失败界面。登陆失败界面如下：



**public** **class** Error1 **implements** IBKmanageView

**public** Error1() {

init();

dealAction();

}

jLabel = **new** JLabel("账号或密码错误",JLabel.***CENTER***);

**（2）主界面**



**（3）各个功能模块**

查询单个食物信息如下：





// 对数据库表中某个对应id值的信息的查找操作

**public** <T> T get(Class<?> klass, Object id) {

**try** {

Connection connection = DataSource.*getConneciton*();

ClassTableDefinition ctd =ClassTableFactory.*getClassTable*(klass);

String SQLSelect = **new** Expression().selectExpressionById(ctd);

PreparedStatement preparedStatement = connection.prepareStatement(SQLSelect);

preparedStatement.setObject(1, id);

ResultSet rs = preparedStatement.executeQuery();

**if** (rs.next()) {

Object obj = klass.newInstance();

ctd.setField(rs, obj);

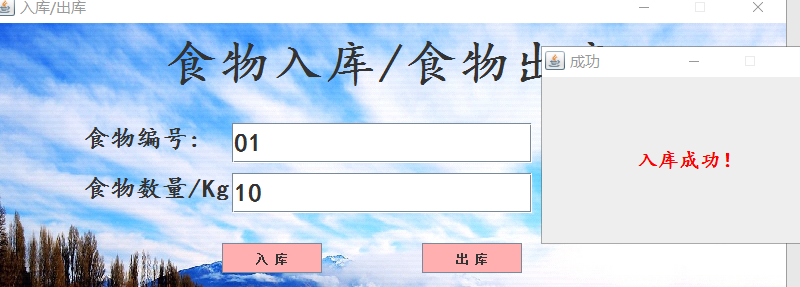
**return** (T) obj;}

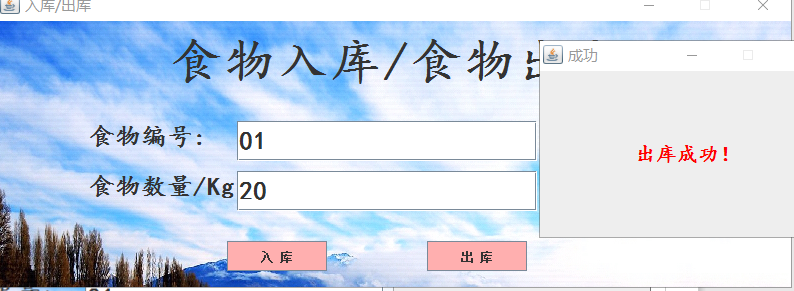
} **catch** (Exception e) {

e.printStackTrace();}

出入库设计如下：







获取当前时间

SimpleDateFormat dateFormat = **new** SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

String Tt = dateFormat.format(data);

Timestamp T = Timestamp.*valueOf*(Tt);

// 对数据库表中增加一条信息的操作

**public** **int** save(Object object) {

Class<?> klass = object.getClass();

ClassTableDefinition ctd = ClassTableFactory.*getClassTable*(klass);

**if** (ctd == **null**) {**return** 0;}

String SQLString = **new** Expression().insertExpression(ctd);

**try** {

Connection connection = DataSource.*getConneciton*();

PreparedStatement state = connection.prepareStatement(SQLString);

Map<String, PropertyColumn> pcMap = ctd.getPorperties();

**int** index = 1;

**for** (String property : pcMap.keySet()) {

PropertyColumn propertyColumn = pcMap.get(property);

state.setObject(index++,propertyColumn.getFieldValue(object));}

**return** state.executeUpdate();

} **catch** (Exception e) {

e.printStackTrace();}

查询所有食物信息界面如下：



// 对数据库表的所有字段的查询操作

**public** <T> List<T> get(Class<?> klass) {

List<T> res = **new** ArrayList<>();

**try** {

Connection connection = DataSource.*getConneciton*();

ClassTableDefinition ctd =ClassTableFactory.*getClassTable*(klass);

String SQLSelect = **new** Expression().baseSelectExpression(ctd);

PreparedStatement preparedStatement = connection.prepareStatement(SQLSelect);

ResultSet rs = preparedStatement.executeQuery();

**while** (rs.next()) {

Object obj = klass.newInstance();

ctd.setField(rs, obj);

res.add((T) obj);

}

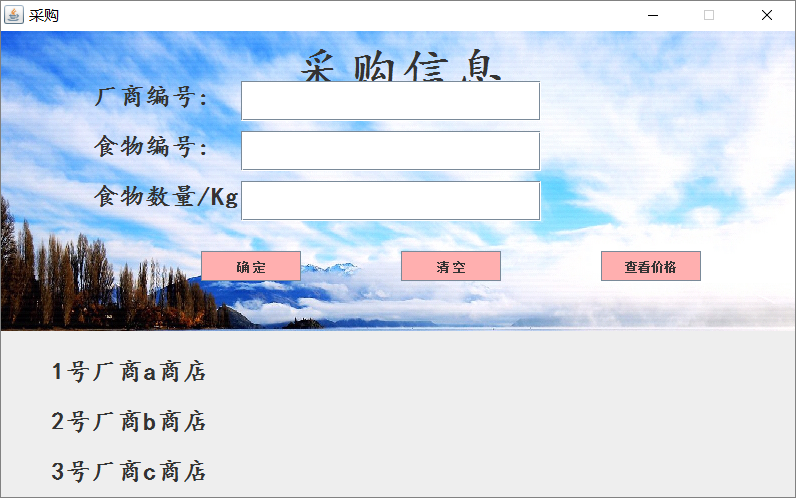
} **catch** (Exception e) {

e.printStackTrace();}

查看所有出入库信息如下：



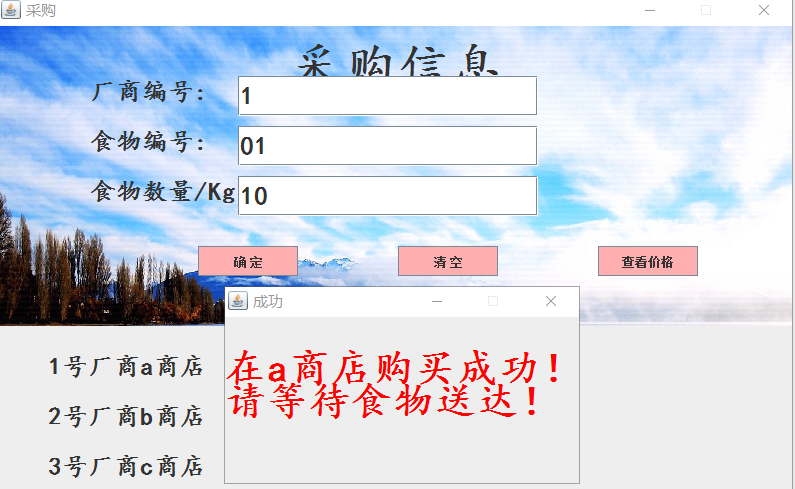
采购界面如下：



查看各商店食物单价信息如下



购买成功页面如下



查看所有采购信息如下：



**7.系统测试**

系统测试是在一个系统或者软件正式投入之前，对系统的功能和质量进行测试，他是系统开发进行到后期的一个重要阶段，能够通过测试来提高系统的质量，减少维护的成本等。本系统针对登陆模块、出入库模块、查询模块、订购模块进行了测试。

1. 登陆模块测试如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试模块 | 测试用例 | 预期效果 | 实际结果 | 是否通过 |
| 登陆模块 | 用户名为admin密码为123123 | 登陆成功 | 登陆成功，跳转主界面 | 通过 |
| 登录模块 | 用户名为admin密码为1111 | 登录失败 | 登录失败，并提示登陆失败 | 通过 |

（2）出入库模块测试

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试模块 | 测试用例 | 预期效果 | 实际结果 | 是否通过 |
| 出入库模块 | 入库  食物编号01  食物数量10 | 入库成功 | 入库成功并提示入库成功 | 通过 |
| 出入库模块 | 出库  食物编号01  食物数量10 | 出库成功 | 入库成功并提示入库成功 | 通过 |

（3）查询模块

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试模块 | 测试用例 | 预期效果 | 实际结果 | 是否通过 |
| 查询模块 | 查询01号食材 | 正确 | 名字数量正确 | 通过 |
| 查询模块 | 查询所有食材 | 正确 | 名字数量正确 | 通过 |
| 查询模块 | 查询所有出入库信息 | 正确 | 名字数量时间正确 | 通过 |
| 查询模块 | 查询采购信息 | 正确 | 名字数量时间正确 | 通过 |

（4）订购模块

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试模块 | 测试用例 | 预期效果 | 实际结果 | 是否通过 |
| 订购模块 | 购买1号商店  食物编号01  食物数量10 | 购买成功 | 购买成功 | 通过 |

通过系统测试，可以看到本系统实现的功能

1. 用户必须输入正确的用户名与密码才能登陆，并对登陆成功与否进行提示。
2. 对食材的出入库进行管理。
3. 对仓库及订购进行查询。
4. 对食材进行订购。
5. **问题分析和总结**

在本次大作业中遇到了很多问题，比如在连接数据库上一开始始终连接不上、java使用mysql语言的困难等等。但是都逐渐通过自己的努力，通过百度，迎难而上，找到自己想要。通过需求分析，概要设计，详细设计等步骤编写程序，修复bug，测试代码，最后将这个智能食堂管理做出来了

在这次的大作业中，让我对书上学到的东西逐渐融会贯通，将知识和时间结合起来。而在这个大作业中也涉及到了数据库的知识，正好帮助我将数据库知识再次复习一遍。通过这次的大作业，让我知道软件工程在实际应用中是十分重要的，我们在做一个项目的时候，要结合实际来确定项目的功能，一个软件从开始到结束需要从许多方面进行考虑，并且需要进行许多遍的修改来达到需求。这次的大作业让我对软件工程这门学科有了更深刻的了解。