# 三、详细设计部分

1. 引言

1.1编写目的

本部分阐明编写详细设计说明书的目的，指名读者对象。

本文档描述每个模块的细节设计，包括模块的接口、调用关系、处理过程和算法，以及模块测试方案等。本文档的主要读者为软件测试人员、模块开发人员、管理人员、测试人员。

## 1.2项目背景

本部分包括项目的来源和主管部门等。

本项目由华中农业大学软件工程学习小组提出，由华中农业大学信息学院2017级计算机科学与技术专业学生完成。

# 1.3定义

本部分列出本文档中所用到的专门术语的定义和缩略词。

本系统只是一个简单的小型养生推荐程序，最终用户为能使用网络且愿意了解水果蔬菜功能性的人群，对教育水平和技术专长没有限制，用户友好性较强，暂无术语分析。

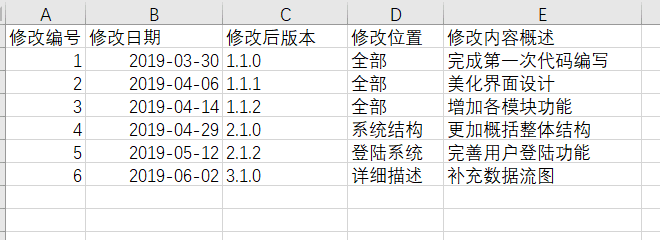
1.4参考资料

[1]详细设计说明书[S].GB 856D-1988.

[2]窦万峰.软件工程方法与实践[M].北京：机械工业出版社，2018.

1.5版本信息

具体版本信息如图表1所示。



图表 1 具体版本信息

1. 总体设计

本部分简要给出系统的需求和软件结构组成，以便追踪到需求和设计。

## 2.1需求概述（定义及描述）

果蔬食客系统的需求定义如下：

2.1.1登录注册子系统

该模块用于用户/管理员注册，登录，找回密码，和游客登录； 用户注册/登录/找回密码方式和管理员注册/登录/找回密码方式有所不同，体现在管理员的注册和登录会有更严格的限制和验证，管理员注册登陆后有权权限更改、注销某用户账号；游客访问仅限于访问公共数据库，只能搜索，预览，但是不能下载资源和评论。

2.1.2用户使用子系统

浏览：浏览网站有关信息

评论：对喜欢的菜品做评论

搜索：快速搜索，转到相关网页

收藏：将某菜单收藏入收藏夹

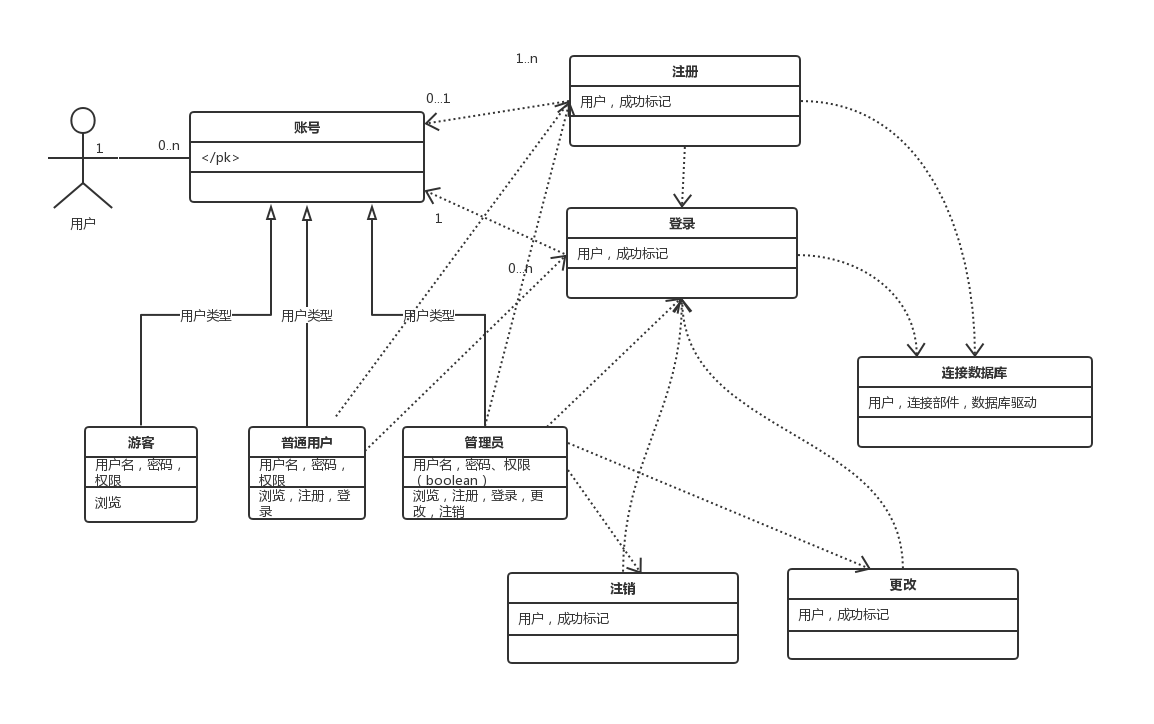
1. 程序描述

## 3.1模块基本信息

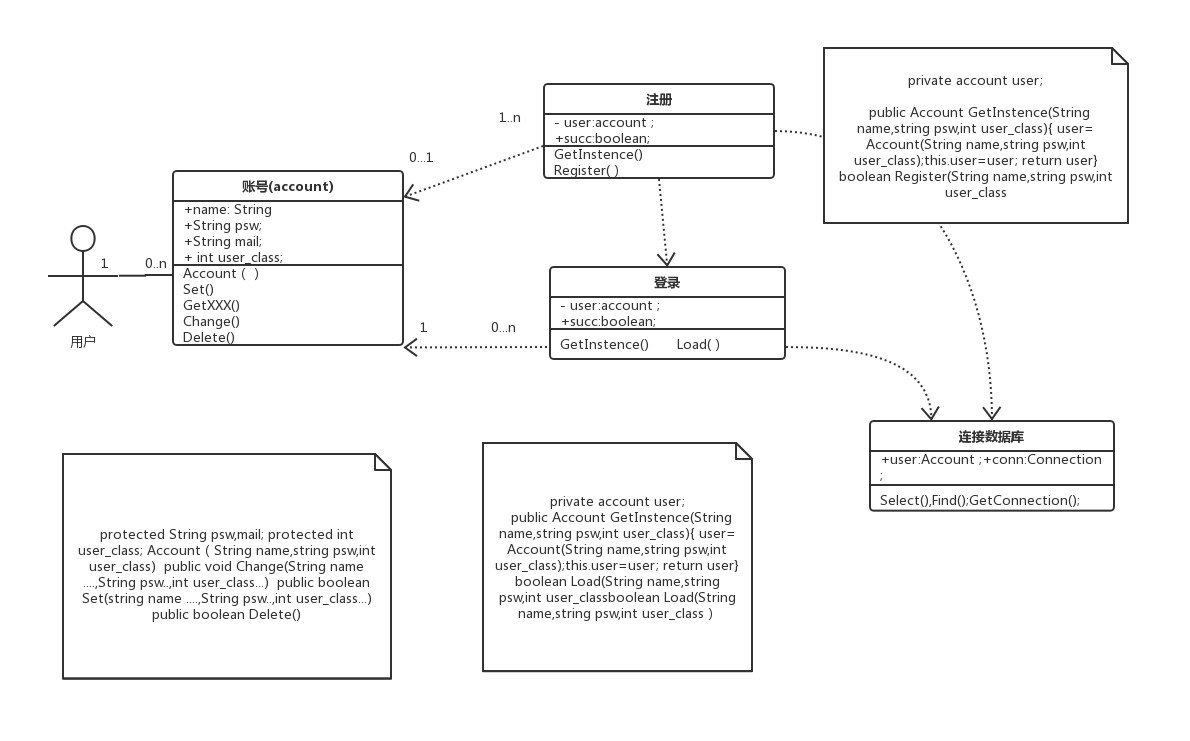
本部分针对每一个模块进行详细设计。

3.1.1登录注册子系统

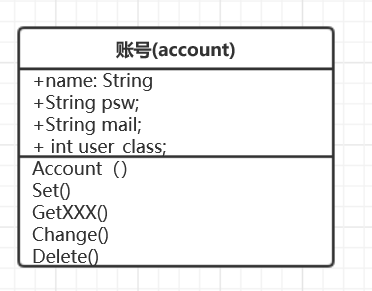
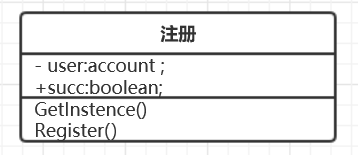
该部分uml类图如下



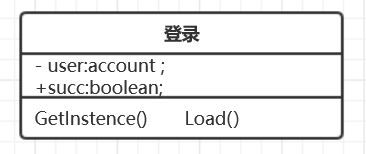
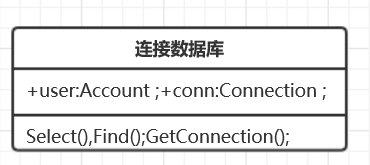
类图1



注册登录精化类图 1

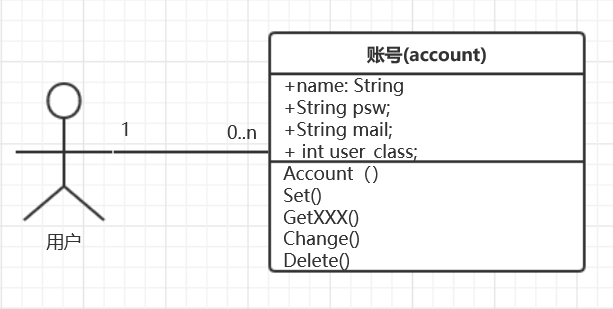
 

类模型\_账号 1 类模型\_注册

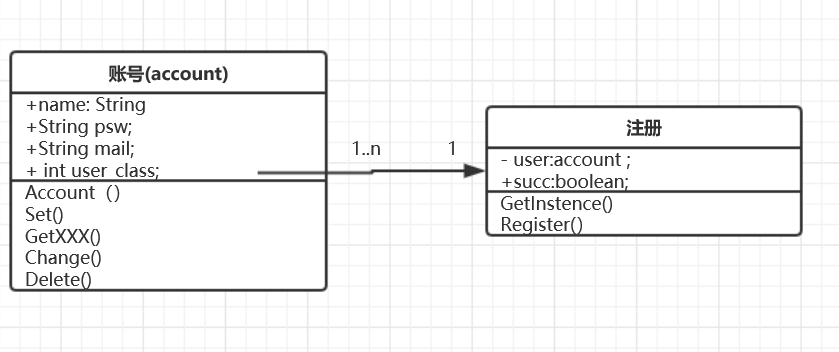
 

类模型\_登录 类模型\_连接

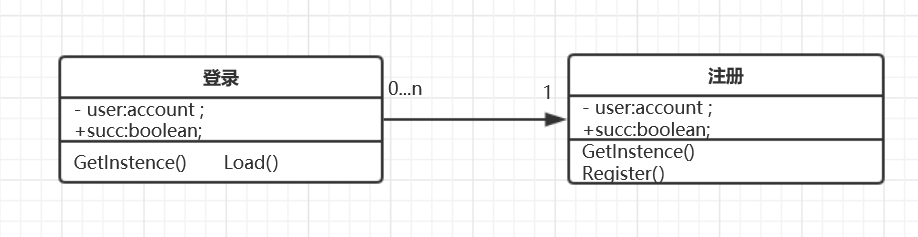
类间关系：



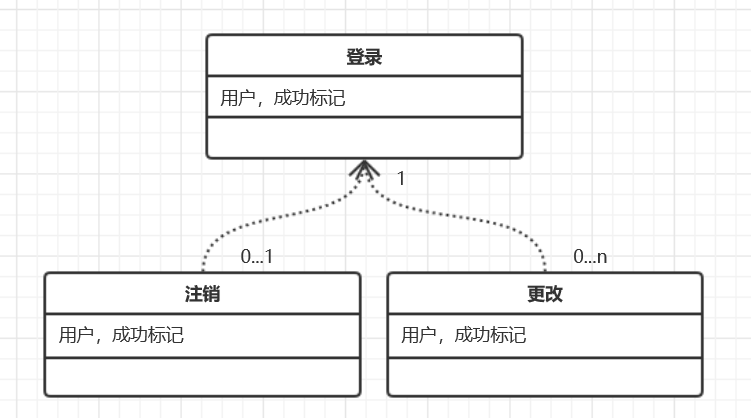
用户——账号



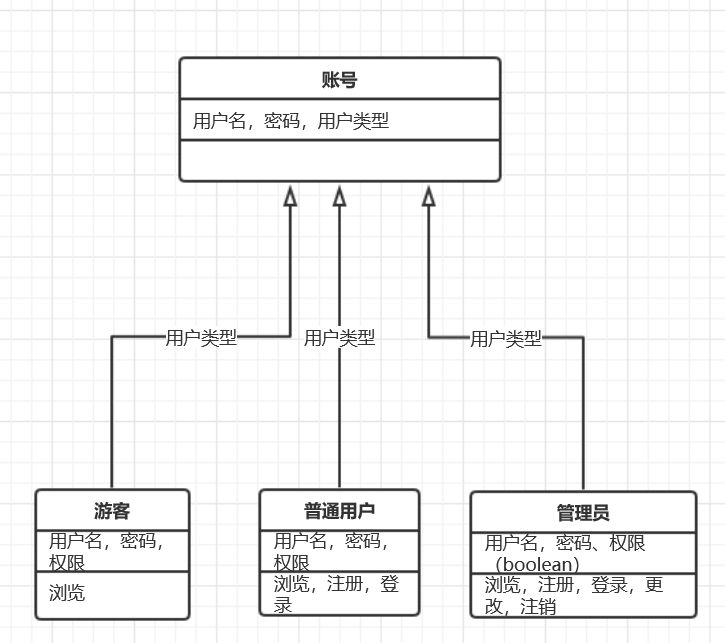
账号——注册



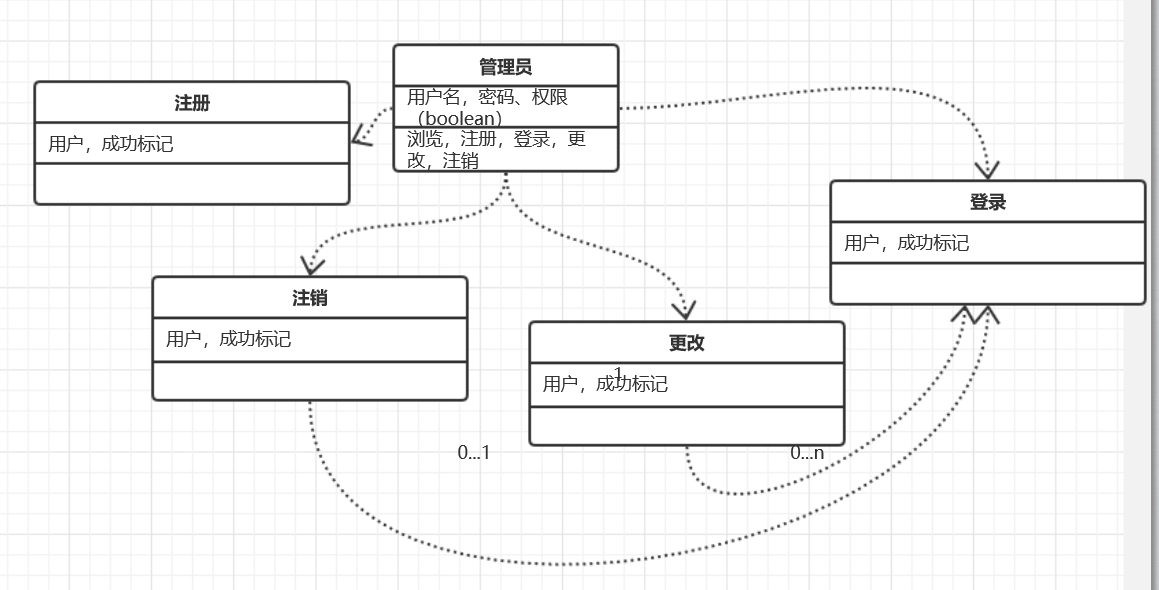
注册——登录



注销、更改——登录



账户类型：账户——派生——游客、普通用户、管理员



类模型\_管理员

程序描述：游客在第一次浏览网页后，点击注册按钮进行注册，如果用户名及密码符合要求，则将用户名及密码保存到数据库中。再次登陆后需要重新输入账号密码，如果和数据库的内容匹配成功则可以登陆。

精度：

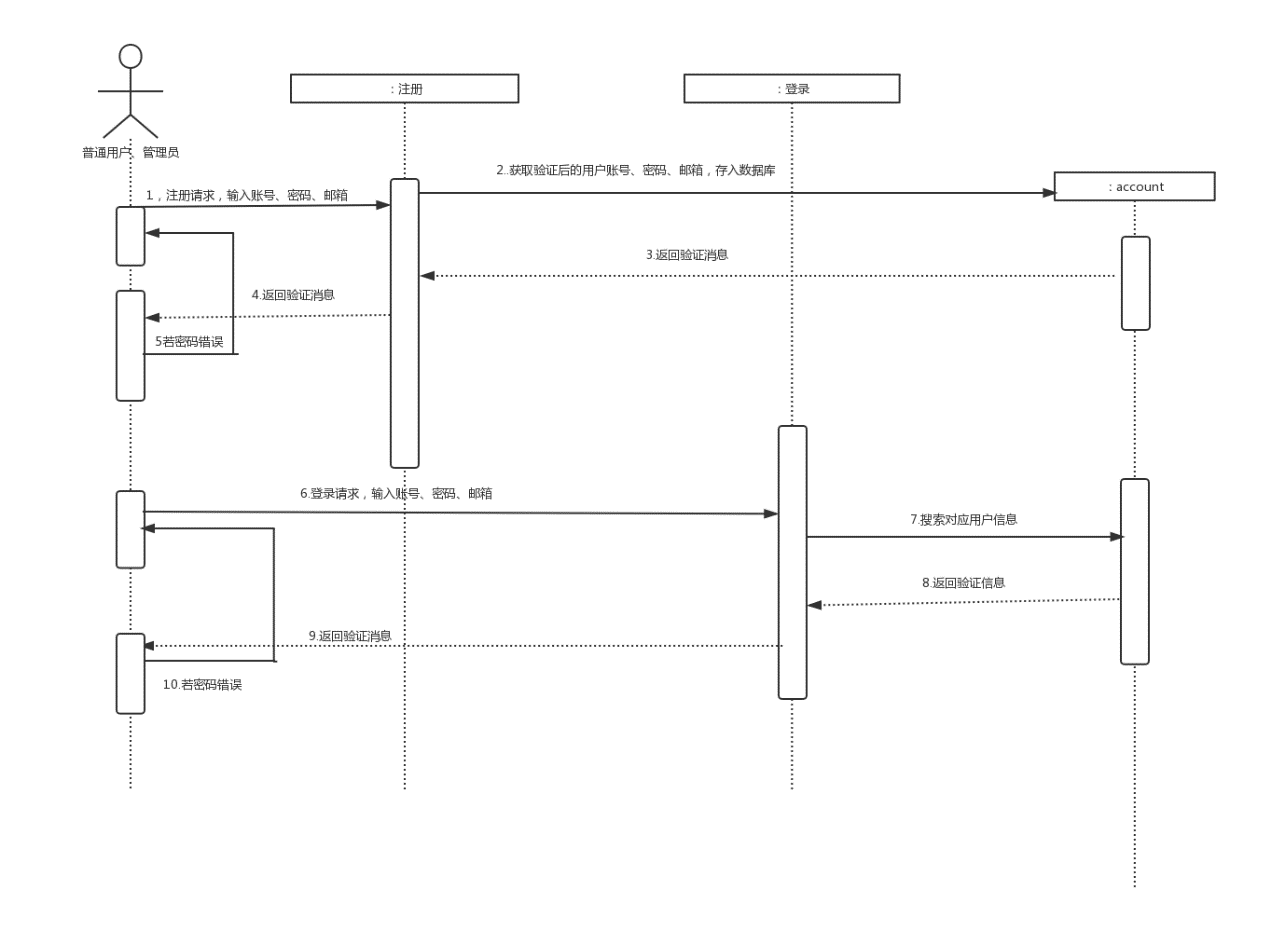
输入数据的精度：字符串数据

输出数据的精度：无输出

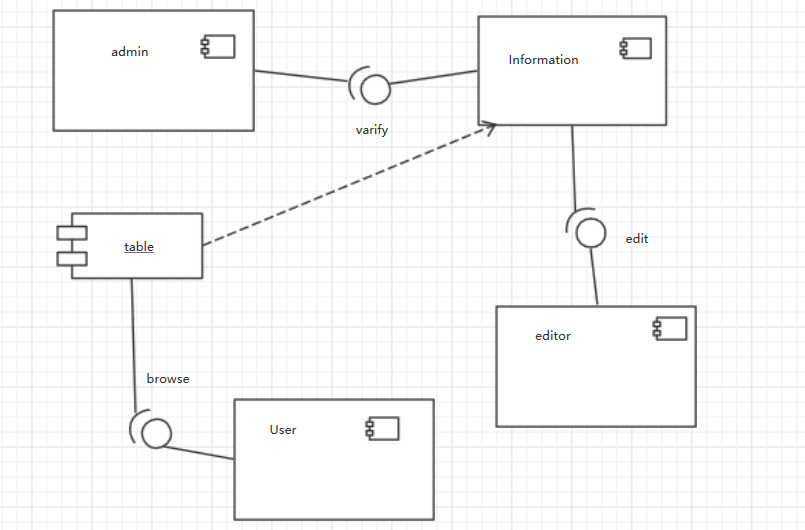
灵活性：   
适用于现在各种操作系统   
时间特性：   
登录：1S以内   
数据处理：0.5s以内   
更新：0.5s以内

算法：通过Java程序写出，首先需要引用Java包中的sql和util包。其次将Java程序与数据库进行相连；注册功能的算法是：输入用户名和密码首先进行对数据库表格id进行比对（util包中方法进行比对），如果没有比对成功，则可以注册，即注册成功；登陆功能的算法是输入用户名和密码和数据库中的用户名和密码进行比对，比对成功即可成功登陆。

具体如下所示：



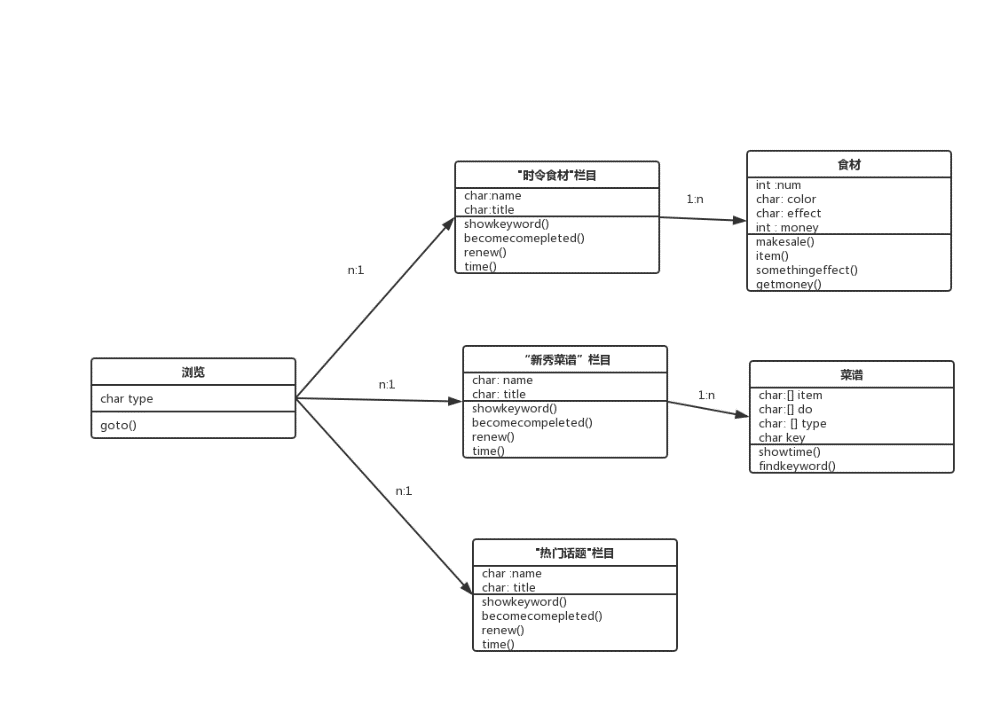
登录顺序图 1



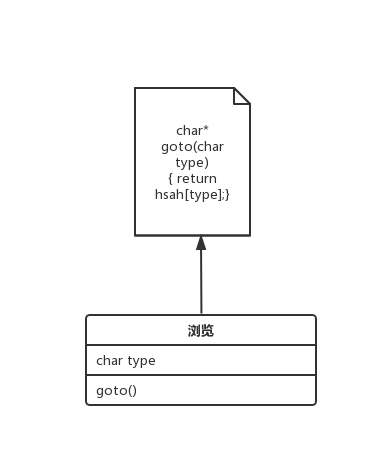
登录注册部署图 1

3.1.2用户使用子系统——浏览模块

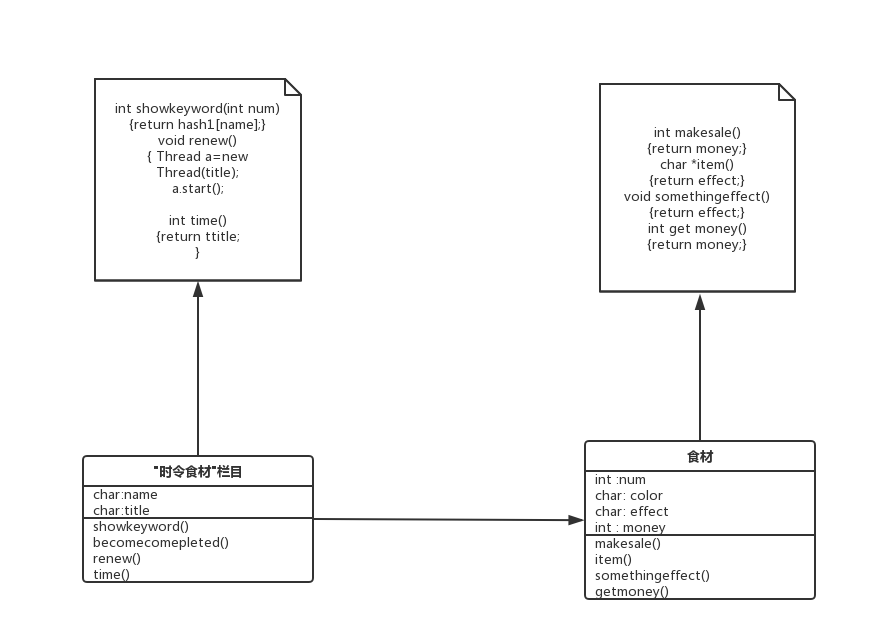
该部分uml类图如下



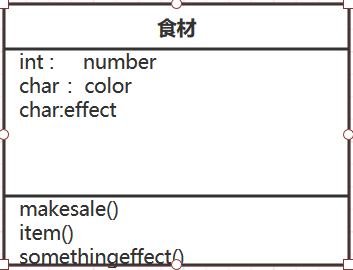
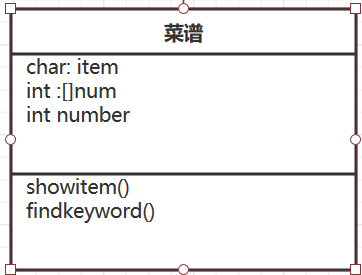
浏览类图 1



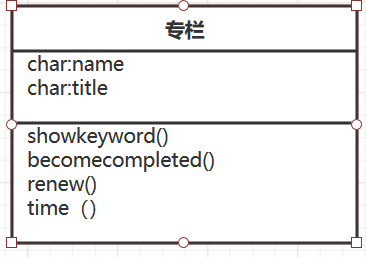
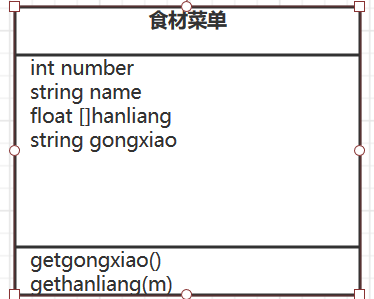
浏览精化类图 1



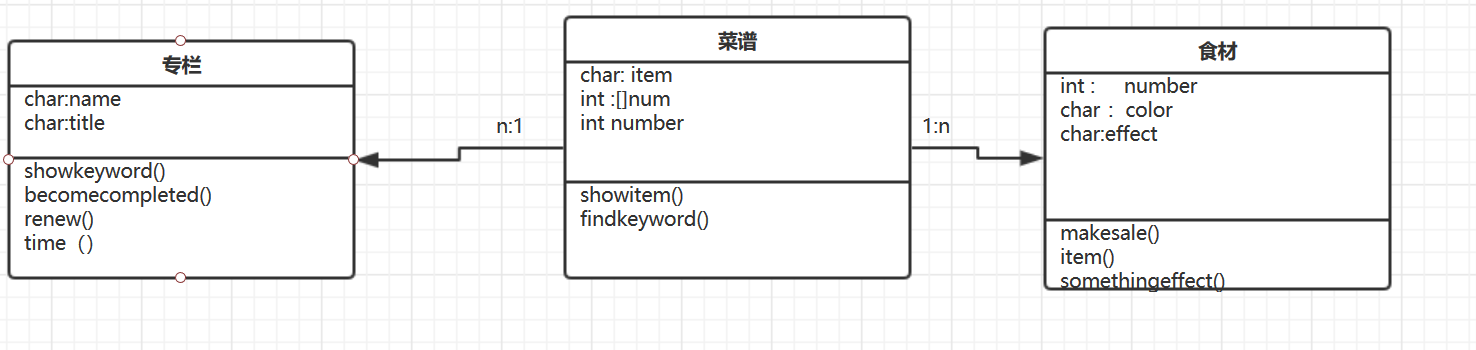
食材精化类图 1

类模型\_食材 类模型\_菜谱

类模型\_话题 类模型\_食材菜单



专栏——菜谱——食材的关系

精度：

输入数据的精度：无精度，进行点击和浏览操作。

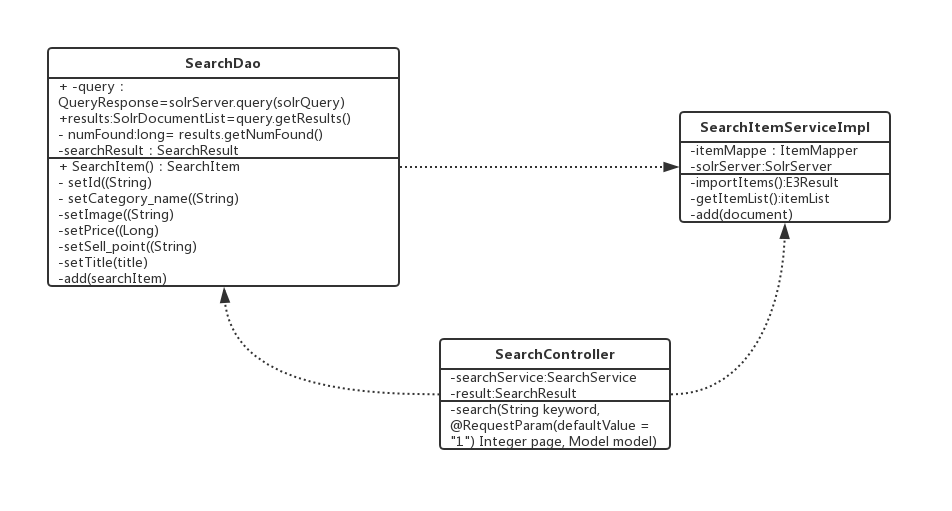
输出数据的精度：无输出

灵活性：   
适用于现在各种操作系统   
时间特性：   
登录：1S以内   
数据处理：0.5s以内   
更新：0.5s以内

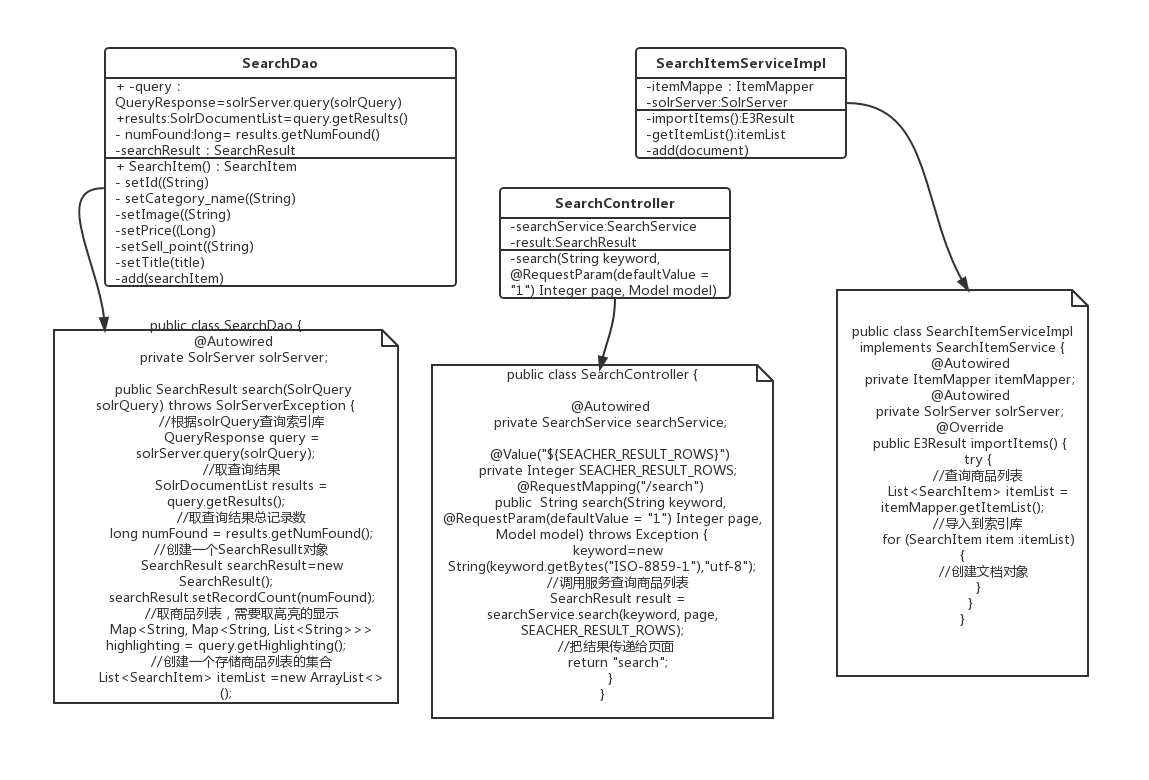
算法：通过相应的点击完成相对应的浏览以及跳转功能。没有比较复杂的算法实现，只需要在html里面写入一些跳转语句即可。

3.1.3用户使用子系统——搜索模块

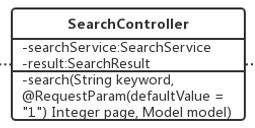
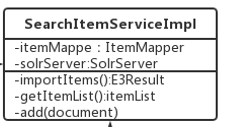
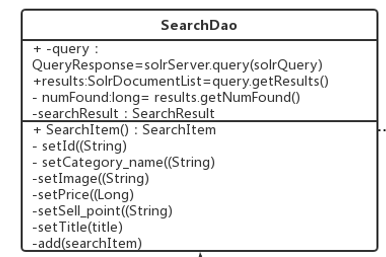
各部分类图如下：



搜索类图 1



搜索模块精化类图 1



类模型

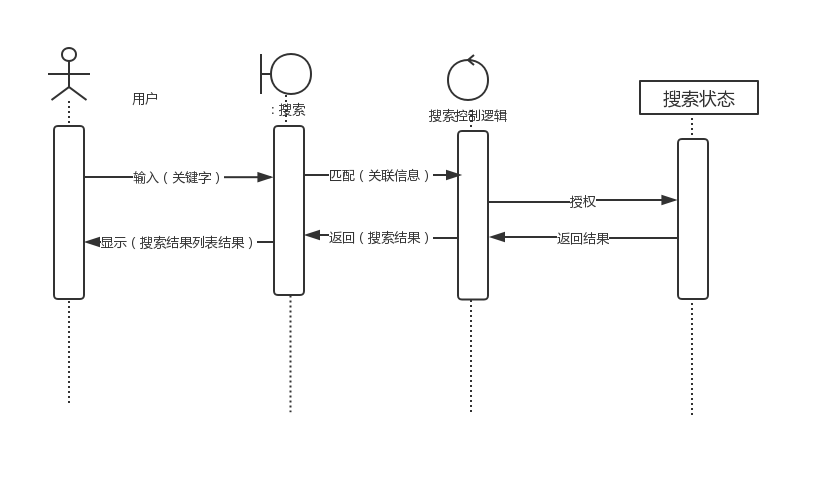
精度：

输入数据的精度：字符串

输出数据的精度：传送字符串的正确度

灵活性：   
适用于现在各种操作系统   
时间特性：   
数据处理：0.5s以内   
更新：0.5s以内

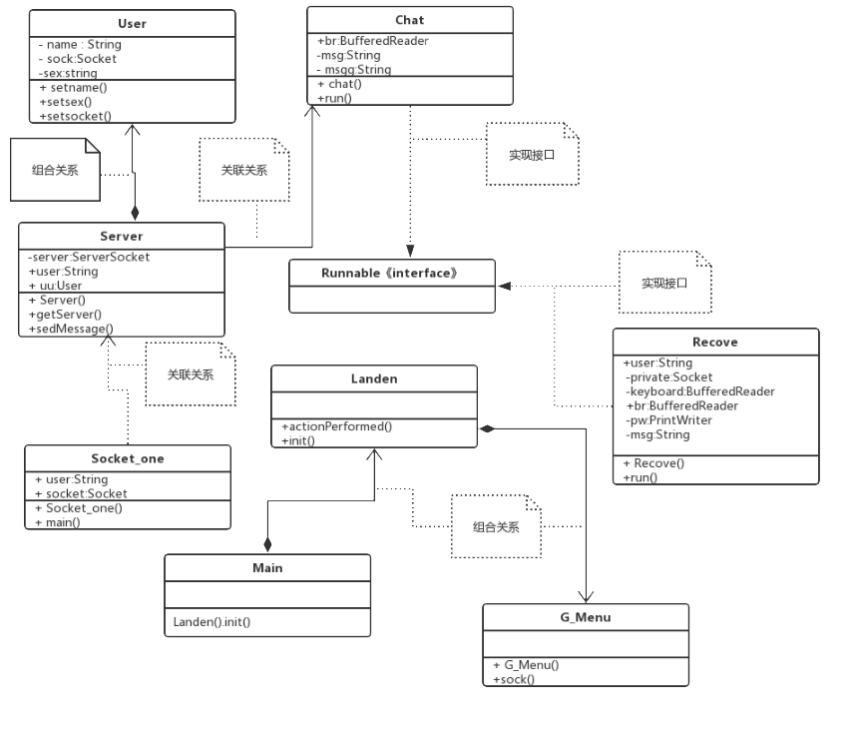
算法：



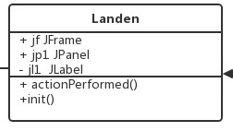
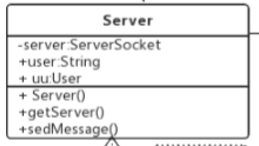
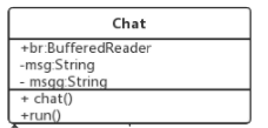
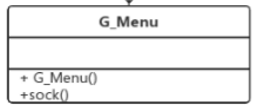
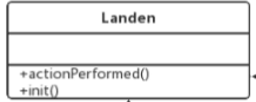
搜索顺序图 1

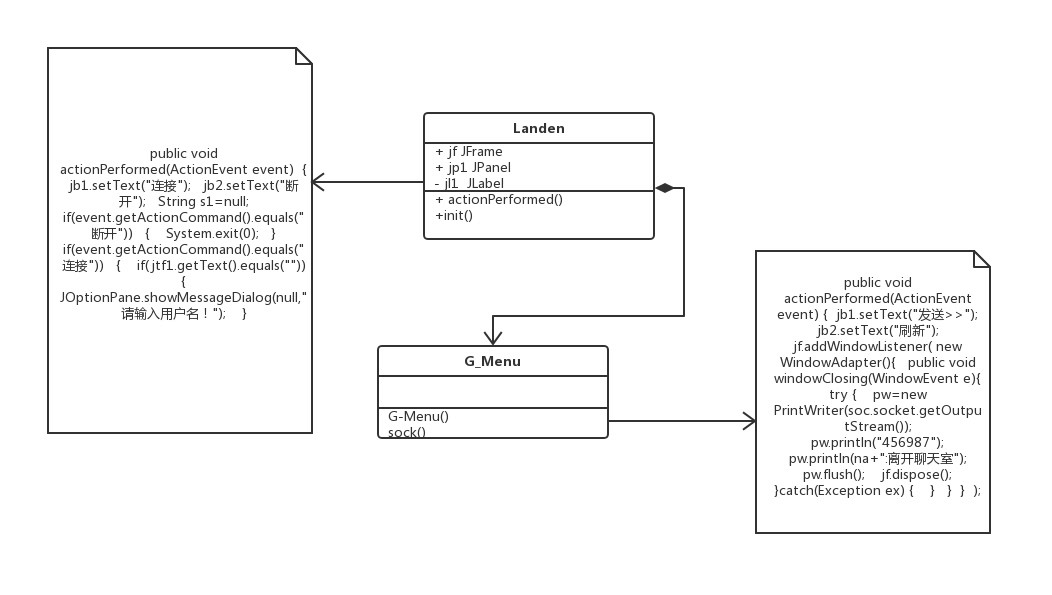
3.1.4用户使用子系统——评论（聊天）模块

各部分精化类图如下：



评论类图 1

类模型 

评论精化类图 1

精度：

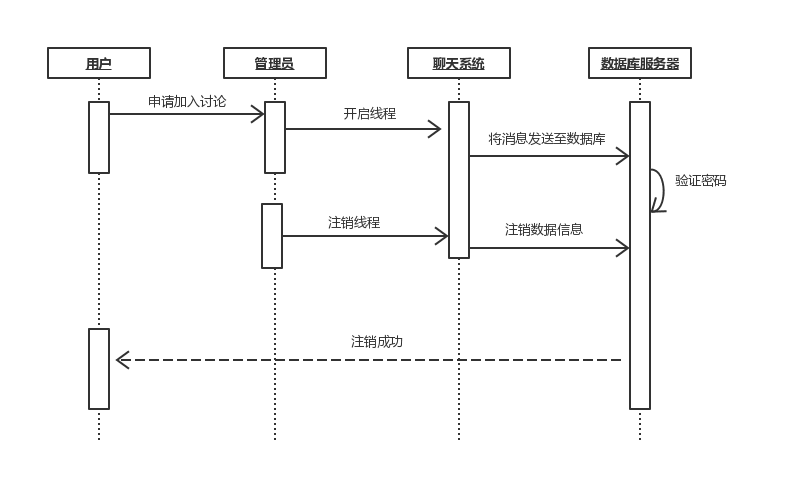
输入数据的精度：字符串或者文件。

输出数据的精度：传送字符串或者文件。

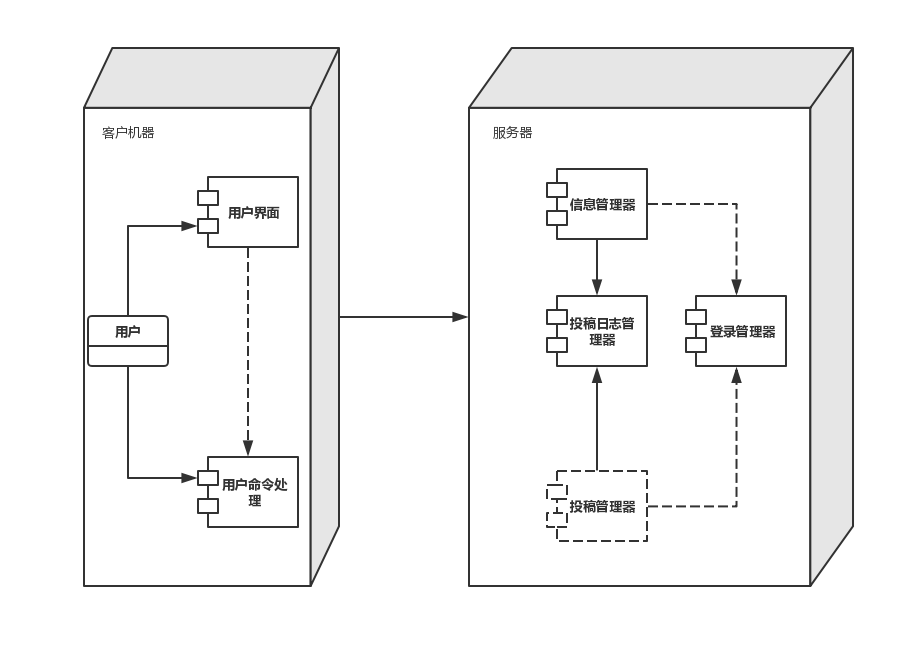
灵活性：   
适用于现在各种操作系统   
时间特性：   
登录：1S以内   
数据处理：0.5s以内   
更新：0.5s以内

算法：

客户端与服务器端建立连接，客户端通过Socket类的构造方法创建一个流套接字并将其连接到指定主机上的指定端口号。服务器端通过ServerSocket类的构造方法实例化ServerSocket对象，并且ServerSocket对象调用accept()方法侦听请求连接指定端口号的该服务器端的客户端，并在接收到客户端请求后返回服务器端的流套接字，即Socket对象，从而服务器端与客户端成功建立连接。建立一个循环监听模式，来一个客户进行一次监听，并创建一个线程。服务器端通过遍历HashMap对象的key值，当找到与userName相等的key值时，利用get()方法取得userName对应的需要私聊的客户端Socket对象，从而将获取私聊的客户端的输出流并将私聊信息进行输出。但是为了客户体验，私聊的客户端需要知道是哪个用户对自己进行私聊操作，故可遍历HashMap对象的value值获取当前客户端的用户名给私聊客户端。群聊信息时，服务器端识别到客户端需要进行的群聊操作，服务器端通过遍历HashMap对象的value值，获取客户端的Socket对象从而利用Socket对象取得输出流后将群聊信息发送给每一个客户端。 客户端的反应与私聊相同。发送信息时，服务器接收到客户端需要进行的传送文件的操作，服务器通过读取文件，并传到相对应的接收着。



聊天室\_时序图 1



评论（聊天）部署图 1

3.3程序逻辑

3.4接口

用户管理端和用户端评论接口。

3.5测试要点

3.5.1登录注册子系统

主要测试以下几种情况：

(1).用户账号重复的情况。

(2).用户账户不存在的情况。

(3).管理员真实身份不正确的情况。

(4).管理员邮箱不存在、失效的情况。

(5)测试管理员权限

3.5.2用户使用子系统

主要测试以下几个要点

昵称重复的情况；

搜索时模糊搜索的精确度；