

南京大学软件工程第十小组

外卖订餐系统

设计说明书



南京大学计算机科学与技术系
Department of Computer Science and Technology Nanjing University

王梓轩、王子威、陈江辉、王锐

2015-3-23

1 引言	3
1.1 编写目的.....	3
1.2 背景.....	3
1.3 定义.....	4
2 总体设计.....	4
2.1 需求规定.....	4
2.1.1 系统功能.....	4
2.1.2 系统性能.....	4
2.2 运行环境.....	5
2.3 基本设计概念和处理流程.....	6
2.4 结构.....	7
2.5 功能需求与程序的关系.....	8
2.6 人工处理过程.....	8
3 接口设计.....	8
3.1 用户接口.....	8
3.2 外部接口.....	9
3.3 内部接口.....	9
4 运行设计.....	10
4.1 运行模块组合.....	10
4.2 运行控制.....	10
4.3 运行时间.....	11
5 系统数据结构设计.....	11
5.1 逻辑结构设计要点.....	11
5.2 物理结构设计要点.....	12
5.3 数据结构与程序的关系.....	14
6 系统出错处理设计.....	14
6.1 出错信息.....	14
6.2 补救措施.....	15
6.3 系统维护设计.....	15

概要设计说明书

1 引言

1.1 编写目的

本说明书是外卖系统的概要设计说明书。

本说明书根据用户提出的需求设计方案编写。将含有以下内容：基本设计概念、运行环境等必要内容，然后对数据接口、软件接口进行了定义，并阐述了物理及逻辑设计的结构要点，并对系统出错状况下出现的问题做出了应对措施。

编写本说明书的目的是：

- a. 为项目开发人员提供依据；
- b. 为项目的测试、升级及维护提供条件；
- c. 为项目负责人布置和控制开发工作提供依据；
- d. 为项目地正常进行提供足够的帮助和有效地指导。

本说明书的预期读者包括：

- a. 项目开发人员；
- b. 项目测试人员；
- c. 项目维护人员；
- d. 技术管理及项目设计人员；
- e. 参与本项目开发进程的其他有关人员。

1.2 背景

随着生活节奏的不断加快，许多人忙于学习或工作而无法有足够的时间出门就餐。也有一些时候，由于天气原因，人们不愿出门。这种情况下，外卖便成为了大家的第一选择。而传统的电话订外卖的形式存在着很多弊端，比如，很多人苦于找不到店家的电话而放弃订餐，再如，电话订餐的效率很低，且没有很好的保障。这种情况下，利用互联网的网上外卖系统便成为了一个好的选择。不仅可以帮助店家进行宣传，也方便食客进行店家的选择，提高了订餐效率，并有了服务保障，使得订餐更加的快捷，方便。此外，我们立足于大学生平台，凭借对大学的熟悉程度及和商家的商谈完成更好的服务。

说明：

- a. 软件名称：打东边来了个喇嘛，打西边来了个哑巴，喇嘛手里拎着五斤挞嘛，哑巴腰里别着个喇叭，别着喇叭的要用喇叭换手里拎着挞嘛的哑巴的挞嘛，拎着挞嘛的哑巴不愿意用挞嘛换手里拎着喇叭的喇嘛的喇叭。拎着喇叭的喇嘛用喇叭打了拎着挞嘛的哑巴，拎着挞嘛的哑巴也用挞嘛打了拎着喇叭的喇嘛外卖订餐系统；
- b. 委托单位：南京大学计算机科学与技术系；

- c. 开发单位：南京大学计算机科学与技术系 2012 级软工小组第 10 组；
- d. 开发团队：组长王梓轩，组员陈江辉、王锐、王子威。

1.3 定义

数据库：存放用户信息及订单信息等相关数据。

2 总体设计

2.1 需求规定

本平台作为一个用于外卖订餐的 Web 应用，提供订餐、留言、在线支付和沟通协商等多方面的功能，让顾客与餐厅能简单有效得进行外卖交易。

2.1.1 系统功能

- a. 顾客
 - 1.注册登录
 - 2.查看菜单
 - 3.编辑/管理订单
 - 4.向餐厅提交订单
 - 5.给餐厅留言
 - 6.在线支付
 - 7.获得和使用红包
 - 8.投诉餐厅
- b. 餐厅
 - 1.注册登录
 - 2.上传/编辑餐厅菜单
 - 3.确认订单
 - 4.查看留言
 - 5.获得餐款
- c. 管理员
 - 1.查看顾客投诉信息
 - 2.管理餐厅、用户账户
 - 3.发放优惠

2.1.2 系统性能

- a. 数据精度
 - 金额以元为单位精确到小数点后两位。

b. 安全性

对用户账号进行加密并且妥善管理，对于数据库信息进行妥善的储存。

c. 稳定性

系统在设计时应注意设计风格统一，使系统可调试性良好，尽可能把系统错误消灭在调试阶段，系统一旦交付使用，就应该有很好的稳定性。

d. 适应性

开发基于的平台要考虑向上兼容性，如操作系统，数据库等要考虑更高版本的兼容性。当需求发生变化时系统应具有一定的适应能力，要求系统能够为将来的变更提供支持。

2.2 运行环境

运行平台：Web

操作系统：Windows XP 及以上

编译环境：PHP

服务器：Apache

数据库：MySQL

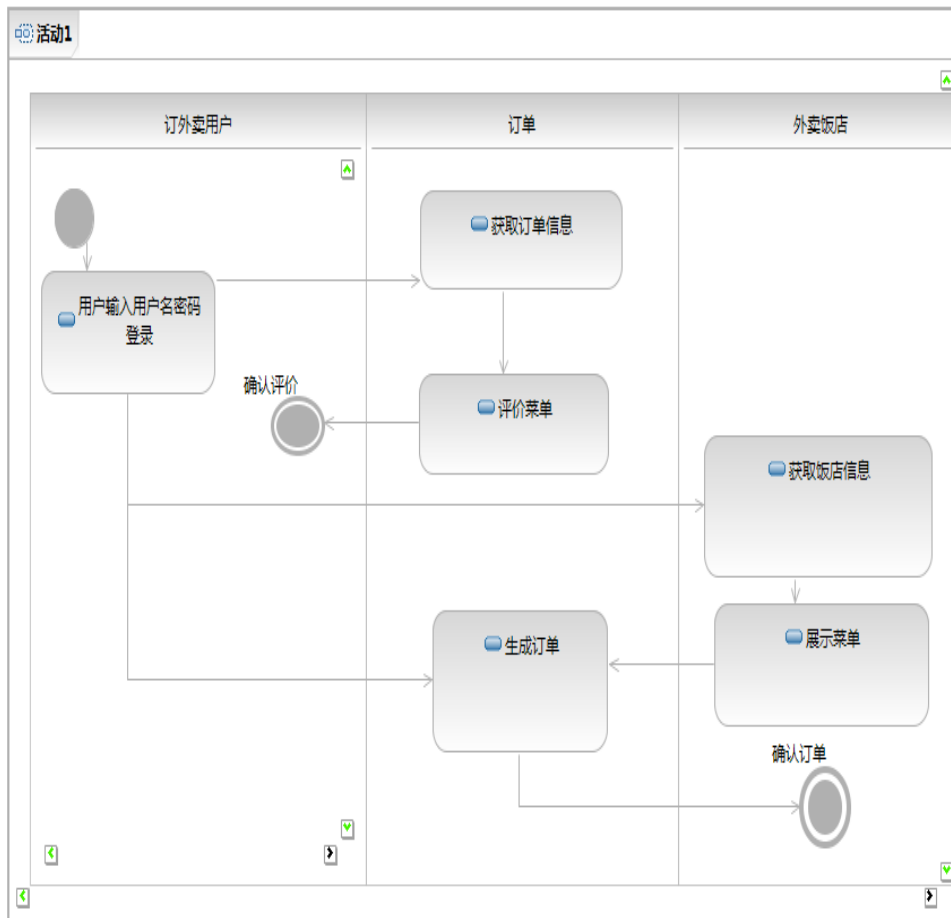
2.3 基本设计概念和处理流程

根据程序的基本运行程序，特将程序的时序图和活动图列下：

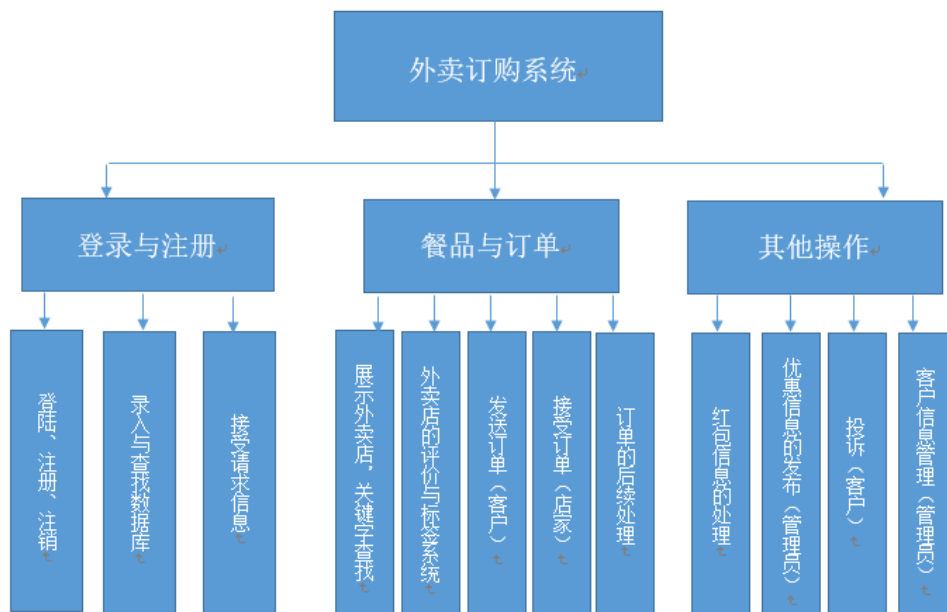
时序图：



活动图：



2.4 结构



2.5 功能需求与程序的关系

功能	输入	功能	输出
用户的注册与登陆			
用户注册	用户信息	录入数据库	注册成功或失败
用户登陆	用户信息	在数据库中查找	登陆成功或失败
用户注销	注销请求	接收请求	返回登陆前状态
查找外卖店			
根据关键字查找	关键字	在数据库中查找	跟关键字相符的店
展示外卖店			
展示外卖店	用户选择外卖店	数据库中取出属性	外卖店的属性
订外卖			
获取订单	用户选择餐点并确定	获得订单	获取订单成功失败
发送订单	收到订单	传输给店家	传输成功或失败
获得订单确认	店家确认	订单存入数据库	存入成功或失败
后续操作			
查看订单	所要查看的订单	在数据库中查找取出	返回订单属性
对订单评价	评价信息	修改数据库	返回新的订单

2.6 人工处理过程

系统管理员需要经常添一些公告，比如近期活动内容的说明等，需要手动改变一些奖励机制与规则，要在必要时进行数据库的维护与恢复，还要根据顾客的投诉信息等来对一些餐厅进行适当惩罚。

3 接口设计

3.1 用户接口

首页界面：包括 3 个部分，分别是登陆部分，提供注册，登陆两个按钮来帮助用户进入注册界面和登陆界面。搜索部分，提供一个文本框输入搜索信息，一个按钮来传输文本框中的信息进入搜索页面。展示部分，在选择框中展示一些排名较高的店家信息，并且点击选择框会进入相应的店家订餐页面。

登录页面，只有 1 个部分，由两个文本框和一个登录按钮组成。用户点击登录，将两个文本框中的内容传输给服务器。如果登录成功返回登录成功页面

注册页面，只有 1 个部分，有多个文本框和一个注册按钮和一个取消注册按钮。点击注册，如果注册成功，返回一个只有文本信息的注册成功页面，在 3s 后返回登录成功页面。如果注册失败，则返回注册页面，并在错误信息边增加文字标注。如果点击取消注册按钮，直接返回首页界面。

登录成功页面，与登录页面基本相同，是 3 个部分，搜索部分与展示部分也跟登录页面

完全相同。而在第一部分不是登录部分，而是一个用户部分，有“用户信息”按钮，点击进入用户信息页面，有“订单”按钮，点击进入用户订单页面，有注销按钮，点击进入首页页面。

用户信息页面，只有 1 个部分，包含用户的大部分信息，和几个按钮，分别是“修改密码”，“修改信息”，“返回”按钮，分别点击进入“修改密码”页面，“修改信息”页面和登陆成功页面。

修改密码页面，有 3 个文本框，分别输入原密码，修改密码和再次输入修改密码，和一个按钮确定。点击确认，如果修改成功返回一个只有文本信息的密码修改成功页面，在 3s 后返回登录成功页面。修改失败则返回修改密码界面，并在错误信息边增加文字标注。

修改信息页面，有多个文本框，和一个按钮确定。点击确认，如果修改成功返回一个只有文本信息的信息修改成功页面，在 3s 后返回登录成功页面。修改失败则返回修改信息界面，并在错误信息边增加文字标注。

用户订单页面，有一个文本标签，里面包含着用户的所有订单信息。和一个返回按钮，点击返回登录成功页面。

搜索页面，跟登录成功页面的前两个部分相同，只不过在展示部分展示的是在搜索中排名较高的店家信息。点击选择框会进入相应的店家订餐页面。

店家订餐页面，第一部分仍是用户部分，第二部分是包含店家信息的店家信息部分，第三部分是菜单部分，由一个多选框和多个文本框和一个确定按钮组成。点击按钮后将多选框，文本框中的信息传输给服务器。

店家订单页面，是一种特殊的用户订单界面，在这个界面中状态为已提交的订单包含一个确认按钮和一个取消按钮，店家点击确认订单状态修改为已接受，店家点击取消，订单状态修改为已拒绝。

即：

用户：登录（），注册（），查看订单（），查看信息（），用户订餐（）；

店家：登录（），注册（），展示菜单（），确认订单（）；

订单：评价订单（），查看订单（），发送订单（）。

3.2 外部接口

外部接口只包含 JDBC 提取信息到服务器的接口，而在数据库中我们现在初步规划有 Customer, Restaurant, Order 三张表需要存储。所以也需要这 3 张表的接口来提取信息。

3.3 内部接口

内部接口只包含 JDBC 提取信息到服务器的接口，而在数据库中我们现在初步规划有 Customer, Restaurant, Order 三张表需要存储。所以也需要这 3 张表的接口来提取信息。

在用户，店家，订单几个类中包括方法用于从这几个类中获取信息和输入信息的方法：

用户：登录（），注册（），查看订单（），查看信息（），用户订餐（）；

店家：登录（），注册（），展示菜单（），确认订单（）；

订单：评价订单（），查看订单（），发送订单（）。

4 运行设计

4.1 运行模块组合

用户注册：输入用户的 ID，密码，以及其他的用户属性。点击确定：建立新的用户对象，调用用户类里的填充属性方法，并存入数据库。点击取消：回到之前的页面。

用户登录：输入用户的 ID，密码，点击登录。在数据库中进行查询，并比对密码，如果相同则返回登录成功，否则返回登录失败。

用户注销：返回首页。

用户查询：从文本框中获得用户查询的信息，在店家数据库中进行查询。调用排名较高的店家类的获取信息函数。将获得的信息按照查询排名从高到低排序添加到选择框中进行输出。

用户选择：从选择框中得到用户点击的选择框的编号，根据编号获得店家。调用店家类的获取信息，获取店家的相关属性并在用户界面显示。

用户点餐：在复选框中进行选择之后点击确定。生成新的订单对象，调用订单类中的函数将复选框中的信息，用户 ID，店家 ID，填充进入该对象。存入数据库，并发送给对应的店家。

店家确认订单：店家收到订单后，点击确认订单，该订单调用修改状态函数。将状态改为“配送中”。

店家取消订单：店家收到订单后，点击取消订单，该订单调用修改状态函数。将状态改为“已取消”。

查看订单：用户可以点击“订单”调取用户类中储存的所有订单，并在选择框中选择某一个。根据选择编号，从数据库中获得相应的订单对象，并获取该订单的具体属性并输出。

评价：用户可以在文本框中输入评价，并点击提交。文本框中的信息会在订单调用的评价函数中输入订单对象。

4.2 运行控制

用户注册：在多个文本框中分别输入用户的 ID，密码，以及其他的用户属性。点击确定或者点击取消。

用户登录：在两个文本框中分别输入用户的 ID，密码，点击登录。

用户注销：点击“注销”，返回首页。

用户查询：用户在文本框中输入查询信息，点击确定。

用户选择：用户在选择框中进行选择。选择框包含该店家的相关信息。

用户点餐：用户在复选框中进行选择之后点击确定。

店家确认订单：店家收到订单后，点击确认订单或取消订单。

查看订单：用户点击“订单”，进入订单页。在该订单页中的选择框点击，则会进入相应订单页。选择框中包含该订单的具体信息。

评价：用户在文本框中输入评价，并点击提交。

4.3 运行时间

数据的传输与接收时间控制在 0.5s 左右，搜索功能的运行时间控制在 1s 左右。

5 系统数据结构设计

5.1 逻辑结构设计要点

名称	标识符	数据项	结构关系
顾客	Customer	id name email pwd photo RedbagList CommentList OrderList	可以索引自己的评论红包以及订单
店家	Saler	id name email photo pwd DishesList OrderList Rate Time	可以索引自己所有的订单以及自家所有菜品
管理员	Administrator	id name email pwd photo	处理一些事务但是自身不需要存储额外的信息
菜品	Dishes	id name saler_id photo price CommentList	从属于店家，可以索引自己的所有评论
评论	Comment	id cus_id dishes_id	

		comment	
订单	Order	id cus_id saler_id price DishesList Confirm Time	用于存储一次订单的信息，维护一个菜品列表
红包	Redbag	price ddl cus_id	
管理记录	AdminRecord	admin_id content	管理员的管理记录

5.2 物理结构设计要点

数据项	说明	访问方法	存取单位	物理关系	设计考虑	保密条件
id	所有的id均为这个单位在同种对象中的唯一标识	数据库	int	数据库存储	同一种结构中不可重复	公开
name	用户名，用于个性化以及识别	数据库	char*	数据库存储	不可重复，支持特殊字符，长度限制	公开
pwd	密码	数据库	char*	数据库存储	长度不得小于6位，不得超过18位	私密，有查看权限
email	邮箱地址，用以提供更好的服务例如作为密码找回的凭证等	数据库	char*	数据库存储	合法的email格式	私密，有查看权限
RedbagList	红包列表，由用户维护	数据库	Redbag*	数据库存储	红包价格有数种	私密，有查看权限
CommentList	评论列表，餐品以及顾客都维护这么一个列表	数据库	Comment*	数据库存储		公开
OrderList	订单列表，店家以及顾客都维护这么一个列表	数据库	Order*	数据库存储		私密，有查看权限
DishesList	餐品列表，由店家维护	数据库	char*	数据库存储		公开

Rate	评级，由用户给店家打分，由店家维护	数据库	enum	数据库存储	有五种评级	公开
saler_id	卖家 id，由菜品和订单维护，用来标识自身所属的店家	数据库	int	数据库存储		私密，仅服务器和管理员可知
cus_id	顾客 id，由评论，订单和红包维护，用来标识自身所属用户	数据库	int	数据库存储		私密，仅服务器和管理员可知
dishes_id	菜品 id，由评论维护，用来标识自身所属菜品	数据库	int	数据库存储		私密
admin_id	由 AdminRecord 维护，用来标识自身所属管理员	数据库	int	数据库存储		公开
price	价值，由菜品，订单和红包维护，用来标识自身的价值	数据库	int	数据库存储		公开
photo	照片，由用户，店家，管理员和菜品维护	数据库	Picture	数据库存储		店家和菜品的公开，用户与管理员的私密，有查看权限
comment	由评价维护，记录评价的内容	数据库	char *	数据库存储		公开
confirm	确认信息，由订单维护，确认这个订单是否是已经送达	数据库	bool	数据库存储		公开
Time	时间，由店家和订单维护，用以记录订单运送时间。	数据库	Time	数据库存储		公开
ddl	截止时间，记录红包使用的截止时间	数据库	Date	数据库存储		公开

5.3 数据结构与程序的关系

	登 录 / 注 册	编 辑 / 信 息	退 出	数 据 / 查 询	编 辑 / 提 交 订 单	发 放 / 使 用 红 包	确 认 / 订 单	取 消 / 订 单	查 看 / 提 交 评 价	发 布 活 动 / 信 息 管 理	添 加 / 更 改 菜 品
Customer	√	√	√	√	√	√		√	√		
Saler	√	√	√	√	√		√	√	√		√
Administrator	√	√	√	√		√		√	√	√	
Dishes				√	√				√		√
Comment				√					√		
Order				√	√	√	√				
Redbag				√		√				√	
AdminRecord				√			√	√		√	

6 系统出错处理设计

6.1 出错信息

序号	出错原因	错误提示及处理方法
1	用户注册时用户名或密码错误。	提示用户用户名或密码错误，用户再次输入用户名和密码。
2	用户登陆时所申请的用户名已被占用。	提示用户所申请的用户名已被占用，用户重新申请其他用户名。
3	不存在消费者搜索的菜品。	返回给消费者错误信息，消费者重新输入查找。
4	消费者定位失败。	提示消费者定位失败，消费者手动选取所在地址。
5	商家的某一菜品已订完但没有及时更新系统，但此时消费者订了该菜品。	向消费者发出提示，让其更换菜品。
6	无法连接到数据库。	页面上显示系统维护，停止订单生成。
7	系统崩溃。	管理员排查故障及恢复。

6.2 补救措施

外卖系统主要是储存人员信息及订单信息，因此需要经常对数据库进行备份，就可以将一些重大错误的损失降到最低。当发生重大错误的时候，我们可以通过日志文件和备份数据来恢复相关的信息。

- a. 我们需要每隔一段时间对数据库进行拷贝，将数据存一份副本，以便在数据库出问题的时候还可以回滚到一段时间之前；
- b. 对于从上一次数据备份之后的操作，需要记录之后对其的更改，这样就可以从备份的数据库中依次执行之后的更改实现数据恢复。

6.3 系统维护设计

因为系统十分依赖数据库的维护，所以我们要确保经常备份数据库，系统就可以安全的运行。

此外，该系统的实时性较强，所以需要我们考虑系统的健壮性；在出错的时候需要及时地进行错误恢复，方能便于用户使用。

而且我们需要使用统一的命名规范和编程习惯，在系统设计上符合软件工程和设计模式的最基本的要求，方便系统的维护、升级和测试。