

**校园找饭**

**软件概要设计说明书**

Bingo



**2013-6-29**

**青岛科技大学**

**吕奎**|**林之梦**|**周星翔**|**李学升**|**靖兆辉**

文档相关信息

项目名称：校园找饭

文档名称：软件概要设计说明书

文档编号：D02

版本号： V1.3

开发团队：Bingo

团队成员：吕奎、林之梦、周星翔、李学升、靖兆辉

所属学校：青岛科技大学

文档修改记录

|  |  |  |  |
| --- | --- | --- | --- |
| 修改人 | 修改时间 | 版本号 | 备注 |
| 吕奎 | 2013年6月29日 | 1.0 | 文档框架及主体内容编写 |
| 吕奎 | 2013年7月21日 | 1.1 | 文档修改及补充，客户端等内容的添加 |
| 吕奎 | 2013年8月17日 | 1.2 | 功能填充 |
| 林之梦 | 2013年9月3日 | 1.3 | 格式修改，版式美化 |

目录

[文档相关信息 1](#_Toc367013609)

[1引言 3](#_Toc367013610)

[1.1编写目的 3](#_Toc367013611)

[1.2背景 3](#_Toc367013612)

[1.3定义 4](#_Toc367013613)

[1.4参考资料 4](#_Toc367013614)

[2总体设计 4](#_Toc367013615)

[2.1需求规定 5](#_Toc367013616)

[2.2运行环境 5](#_Toc367013617)

[2.2.1 总体开发环境 5](#_Toc367013618)

[2.2.2 WebService端开发环境 5](#_Toc367013618)

[2.3基本设计概念和处理流程 6](#_Toc367013619)

[2.4结构 7](#_Toc367013620)

[2.5功能器求与程序的关系 7](#_Toc367013621)

[2.6人工处理过程 7](#_Toc367013622)

[2.7尚未问决的问题 7](#_Toc367013623)

[3接口设计 7](#_Toc367013624)

[3.1用户接口 7](#_Toc367013625)

[3.2外部接口 8](#_Toc367013626)

[3.3内部接口 8](#_Toc367013627)

[4运行设计 8](#_Toc367013628)

[4.1运行模块组合 8](#_Toc367013629)

[4.2运行控制 9](#_Toc367013630)

[4.3运行时间 9](#_Toc367013631)

[5系统数据结构设计 9](#_Toc367013632)

[5.1逻辑结构设计要点 9](#_Toc367013633)

[5.2物理结构设计要点 9](#_Toc367013634)

[5.3数据结构与程序的关系 9](#_Toc367013635)

[6系统出错处理设计 10](#_Toc367013636)

[6.1出错信息 10](#_Toc367013637)

[6.2补救措施 10](#_Toc367013638)

[6.3系统维护设计 10](#_Toc367013639)

**软件概要设计说明书**

**1引言**

**1.1编写目的**

该阶段要把软件需求转化为软件表示，即描绘出软件的总体概貌。完成包括程序系统的基本处理流程、程序系统的组织结构、模块划分、功能分配、接口设计、运行设计、数据结构设计和出错处理设计等，为程序的详细设计提供基础。

**1.2背景**

**说明：**

a.待开发软件系统的名称；

b.列出此项目的任务提出者、开发者、用户以及将运行该软件的计算站（中心）。

##### 系统名称：校园找饭

##### 任务提出者：**Bingo团队**

##### 开发者：Bingo团队

##### 面向用户：提供餐饮服务的商家和大学生消费者及其他需要此方面**服务的消费者**

##### 特此声明：**此软件的开发作为后期商业项目的原型，均采用免费开发**工具，开发全部由Bingo团队完成，未经允许不得随意发布和使用。

**1.3定义**

* **Web 服务：**Web服务是在互联网上的一种服务协议，采用XML作为基本的标记语言，SOAP作为互操作协议，WSDL作为服务描述语言，通过UDDI可以把服务注册到互联网以便搜索，为更多的用户使用其服务。不同公司及组织对其定义各不相同，但有几点是一致的。首先，它是由企业驱动和应用驱动而产生的；其次，它具有分布性、松散藕合、可复用性、开放性以及可交互性等特性。
* **WebService：**Web Service是一项新技术， 能使得运行在不同机器上的不同应用无须借助附加的、专门的第三方软件或硬件， 就可相互交换数据或集成。 Web Service减少了应用接口的花费。Web Service为整个企业甚至多个组织之间的业务流程的集成提供了一个通用机制。
* **SOAP：**Simple Object Access Protocol。
* **WSDL：**Web Service Description Language。

**1.4参考资料**

1. 需求说明书模版GB 8567-88。
2. 大赛题目设置：《基于Web Service的云端应用软件开发》，详情可参见<www.softqilu.com>。
3. 参考书籍：

（1）《软件工程与软件文档写作》 文斌，刘长青，田原。 北京：清华大学出版社；北京交通大学出版社，2006.8；

4. 其它相关文档：

1. 《概要设计说明书》
2. 《详细设计说明书》
3. 《用户手册》

**2总体设计**

**2.1需求规定**

说明对本系统的主要的输入输出项目、处理的功能性能要求，详细的说明可参见附录C。

**2.2运行环境**

### 2.2.1 总体开发环境

#### 2.2.1.1 硬件环境

* 网络设备：网线若干米、交换机、路由器；
* 畅通的网络环境；
* 有条件的话,一部Android平台的手机，系统采用Android2.3以上操作系统。

#### 2.2.1.2 软件环境

* 操作系统：Windows7操作系统；
* WebService服务器：Tomcat7.0
* 开发工具：Eclipse IDE for Java EE Developer，JDK,JDBC

### 2.2.2 WebService端开发环境

#### 2.2.2.1 硬件环境

* CPU：至少双核CPU，1.2GHz以上主频；
* 内存：至少2G；
* 硬盘：至少320G；
* 网络：至少500KB/s的网速
* 其他外设：无特别要求。

#### 2.2.2.2 软件环境

* 操作系统：Windows 7；
* 数据库管理系统：Mysql5.0，Navicat for Mysql
* 开发工具:Eclipse IDE for Java EE Developer，jdk-7u25,Notepad++

**2.3基本设计概念和处理流程**

 消费者通过校园找饭平台，通过软件的自动定位或者手动选择学校后看到自己附近的餐馆信息，若想进行订餐服务，首先需要注册成为我们的会员。注册成功并且登陆后就可以对自己看好的饭菜进行下单，下单完成后平台后台有推送功能，商家会马上收到并处理订单信息，此时消费者也会收到推送消息，提示商家已经确认订单，这样就避免了只提交订单却无人应答的尴尬。在商家送饭到消费者的手中后，商家会把订单状态置为已送达，消费者消费完后也需要确认状态，并对此个订单做出评价。

**2.4结构**

用一览表及框图的形式说明本系统的系统元素（各层模块、子程序、公用程序等）的划分，扼要说明每个系统元素的标识符和功能，分层次地给出各元素之间的控制与被控制关系.

**2.5人工处理过程**

本软件有运行在消费者和商家之间，工作过程中不再需要其他人工干预。

**2.6尚未问决的问题**

对软件的并发性控制尚没做明确的测试。

**3接口设计**

**3.1用户接口**

在用户界面部分，根据需求分析的结果，用户需要一个用户友善、简单界面。在界面设计上，应做到简单明了，易于操作，并且要注意到界面的布局，应突出的显示重要以及出错信息。外观上也要做到合理化，考虑到用户对淘宝、美团等app风格较熟悉，所以我们在开发的过程也参考了一下它们的布局。在设计语言上，已决定使用Java 进行编程，在界面上可使用Android所提供的可视化组件。 其中服务器程序界面要做到操作简单，易于管理。管理者在管理数据库的过程中，也可以采用辅助工具，如Navicat foy mysql

总的来说，系统的用户界面应作到可靠性、简单性、易学习和使用,并且努力具有自己的特色。

**3.2外部接口**

#### 3.1.1软件接口

服务器程序可使用JDBC连接数据库，进行对数据库的所有访问。

Anroid手机客户端通过调用webservices提供的方法，实现对数据库的访问或操作。

#### 3.1.2硬件接口

在输入方面，对于键盘、鼠标的输入，可用Java的标准输入/输出，对输入进行处理。

在输出方面，通过手机屏幕对内容进行输出

**3.3内部接口**

内部接口方面，各模块之间采用函数调用、参数传递、返回值的方式进行信息传递。具体参数的结构将在下面数据结构设计的内容中说明。接口传递的信息将是以数据结构封装了的数据，以参数传递或返回值的形式在各模块间传输。

**4运行设计**

**4.1运行模块组合**

说明对系统施加不同的外界运行控制时所引起的各种不同的运行模块组合，说明每种运行所历经的内部模块和支持软件。

客户端在有输入时启动接收数据模块，通过各模块之间的调用，读入并对输入进行JSON包装。通过调用WebService，在服务器端对数据进行处理，返回需要的数据到客户端，客户端再对信息进行处理，产生相应的输出。

服务器程序的接收网络数据模块必须始终处于活动状态。接收到数据后，调用数据处理/查询模块对数据库进行访问，处理后对数据进行包装发送到客户端。

**4.2运行控制**

运行控制将严格按照WebServices函数调用关系来实现。在各事务中心模块中，需对运行控制进行正确的判断，选择正确的运行控制路径。

在网络传输方面，客户机在发送数据后，将等待服务器的确认收到信号，收到后，再次等待服务器发送回答数据，然后对数据进行确认。服务器在接到数据后发送确认信号，在对数据处理、访问数据库后，将返回信息送回客户机，并等待确认。

**4.3运行时间**

在软体的需求分析中，对运行时间的要求为必须对作出的操作有较快的反应。网络硬件对运行时间有最大的影响，当网络负载量大时，对操作反应将受到很大的影响。所以建议采用硬件配置较好的计算机模拟服务端，实现客户机与服务器之间的连接。

**5系统数据结构设计**

**5.1逻辑结构设计要点**

将结构概念转化为某个DBMS支持的数据模型进行优化，根据ER图设计关系模型，形成数据库逻辑模式，建立视图形成外模式

数据库设计

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名 | 消费者信息表（consumerinfo） | | | | |
| 字段 | 类型 | 长度 | 允许为空 | 主键 | 说明 |
| Conname | Varchar | 11 | No | Yes | 消费者账号 |
| Conpassword | Varchar | 32 | No | No | 消费者密码 |
| Conaddress | Varchar | 64 | No | No | 消费者地址 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名 | 菜品详细信息表Dish | | | | |
| 字段 | 类型 | 长度 | 允许为空 | 主键 | 说明 |
| Disid | Int | 64 | No | Yes | 菜品的id，需要自动增长 |
| Dianame | Varchar | 16 | No | Yes | 菜品名称 |
| Disprice | Int | 4 | No | Yes | 菜品价格 |
| Disdesc | Varchar | 64 | No | Yes | 菜品简单描述 |
| Disphoto | Varchar | 100 | No | Yes | 菜品的图片编号 |
| Disseller | Varchar | 20 | No | No | 菜品的销售者 |
| Stop | Int | 1 | No | No | 默认值为0，代表商品正在销售 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 表名 | | 订单详细信息表orderinfo | | | | | |
| 字段 | 类型 | | 长度 | 允许为空 | 主键 | 说明 |
| Orderid | Int | | 11 | No | Yes | 订单的id，需要自动增长 |
| Ogood | Varchar | | 64 | No | No | 订单的菜品的名称 |
| Oprice | Int | | 5 | No | No | 菜品的价格 |
| Otime | Date | | 0 | No | No | 记录订单生成的时间 |
| Ostatus | Int | | 1 | No | No | 订单的状态，订单生成时默认为0，代表订单已生成但商家未确认 |
| Oreason | Varchar | | 64 | Yes | No | 用户下单使得备注，默认允许为空 |
| Conname | Varchar | | 20 | No | No | 用户下单时保存用户的账号 |
| Seller | Varchar | | 20 | No | No | 此订单的菜品的商家 |
| Address | Varchar | | 64 | No | No | 用户下单时需要送达的地址 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表名 | 商家详细信息表sellerinfo | | | | |
| 字段 | 类型 | 长度 | 允许为空 | 主键 | 说明 |
| Seller | Varchar | 20 | No | Yes | 商家的账号 |
| Selpassword | Varchar | 12 | No | No | 商家的密码 |
| Selphone | Varcahr | 32 | No | No | 商家的电话 |
| Resname | Varchar | 64 | No | No | 商家的餐馆名称 |
| Nearschool | Varchar | 64 | No | No | 商家附近的学校 |
| Seladdress | Varchar | 128 | No | No | 商家的地址 |
| Seldesc | Varchar | 128 | No | No | 商家的简单描述 |

**5.2物理结构设计要点**

给出本系统内所使用的每个数据结构中的每个数据项的存储要求，访问方法、存取单位、存取的物理关系（索引、设备、存储区域）、设计考虑和保密条件。

**5.3数据结构与程序的关系**

说明各个数据结构与访问这些数据结构的形式:

**6系统出错处理设计**

**6.1出错信息**

程序在运行时主要会出现两种错误：1、由于输入信息，或无法满足要求时产生的错误，称为软错误。2、由于其他问题，如网络传输超时等，产生的问题，称为硬错误。

对于软错误，须在各种验证模块由数据进行数据分析，尽量对用户进行必要的提示以减少软错误出现的次数。

对与硬错误，在代码中都在必要的地方使用了“异常机制”进行处理，一般分析异常就可以及时处理出现的问题。

**6.2补救措施**

系统主要分为两部分：数据库、服务器和手机客户端。

对于手机客户端，出现的问题一般对用户都有相应的提示，并且我们还有反馈功能，用户可将崩溃的信息直接调用手机邮件功能发送给我们，以便我们对错误进行统计分析。

对于数据库、服务器端，我们建议采用可靠、稳定的服务器机种，保证系统运行时的可靠性。数据库可通过MySQL的日志文件，对其进行ROLLBACK 处理，对数据进行恢复，并且建议在使用过程中，适时对数据库数据进行备份。

**6.3系统维护设计**

维护方面主要为对服务器上的数据库数据进行维护。可使用 MySQL 的数据库维护功能机制。例如，定期为数据库进行Backup，维护管理数据库死锁问题和维护数据库内数据的一致性等。