|  |
| --- |
|  |
| 懒人请投币 |
| 概要设计说明书 |

版本号： V1.10

**修订历史记录**

| **日期** | **版本** | **说明** | **作者** |
| --- | --- | --- | --- |
| 2015/9/20 | 1.0 | 建立文档 | 黄一璐，熊振平，吕加 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[**修订历史记录** ii](#_Toc425321766)

[**1** **概述** 1](#_Toc425321767)

[1.1 系统概述 1](#_Toc425321768)

[1.2 文档概述 1](#_Toc425321769)

[2 设计方法和原则 1](#_Toc425321770)

[2.1 限制和约束 1](#_Toc425321771)

[2.1.1 开发技术 1](#_Toc425321772)

[2.1.2 开发环境 1](#_Toc425321773)

[2.1.3 计划时间 1](#_Toc425321774)

[2.2 设计遵循原则 1](#_Toc425321775)

[2.3 设计方法 2](#_Toc425321776)

[3 总体设计 2](#_Toc425321777)

[3.1 模块划分 2](#_Toc425321778)

[3.2 模块调用关系 3](#_Toc425321779)

[4 数据库设计 3](#_Toc425321780)

[**4.1** 外部设计 3](#_Toc425321781)

[4.1.1 标识符 3](#_Toc425321782)

[**4.2** **支持软件** 3](#_Toc425321783)

[4.2.1 数据库命名规则 3](#_Toc425321784)

[4.2.2 数据库对象命名规则 3](#_Toc425321785)

[4.2.3 字段命名规则 4](#_Toc425321786)

[4.3 数据库的逻辑结构设计 4](#_Toc425321787)

[4.4 数据库的物理设计 4](#_Toc425321788)

[4.4.1 表汇总 5](#_Toc425321789)

[4.4.2 视图设计 7](#_Toc425321790)

[4.4.3 存储过程设计 7](#_Toc425321791)

[4.4.4 触发器设计 8](#_Toc425321792)

[4.5 安全性设计 8](#_Toc425321793)

[4.5.1 防止用户直接操作数据库的方法 8](#_Toc425321794)

[4.5.2 用户账号密码的加密方法 8](#_Toc425321795)

[4.6 数据库管理与维护说明 8](#_Toc425321796)

[5 出错和维护设计 8](#_Toc425321797)

[5.1 出错处理 8](#_Toc425321798)

[5.1.1 出错信息 8](#_Toc425321799)

[5.1.2 扑救措施 9](#_Toc425321800)

# **概述**

系统概述

本系统全称是懒人请投币即生活服务类软件。

软件是利用服务器+MySQL进行管理数据，效率高，速度快，对用户资料进行备份还原，对用户个人信息进行储存。

文档概述

该文档是为编程人员提供设计方法、系统接口和数据库模型。并对软件详细设计和编码提供重要依据。

文档包括：设计方法，系统接口，数据库模型。

设计方法和原则

本系统架构采用MVC模式来实现。

限制和约束

### 开发技术

开发语言：Objective\_C，PHP

数据库：MySQL，sqlite

其他技术：JSON，HTML，ThinkPHP

### 开发环境

操作系统：MAC OS 10.10.4

开发工具：Xcode7.0

数据库：MySQL

运行环境：iOS9.0

### 计划时间

计划时间：5周；详细计划时间请参见【项目开发计划】。

设计遵循原则

1. 命名规则：意义明确无二义、简洁、无重复
2. 模块独立性原则：避免重复定义，处理完整，输入输出明确
3. 边界设计原则：包括外部接口、内部接口和用户接口
4. 数据库：最低满足3NF
5. 安全和保密原则：数据安全，可靠
6. 系统的灵活性：适用性及可维护性
7. 用户易用性原则：操作方便，功能完善

设计方法

为了实现系统的高效性，稳定性，易维护新，采用MVC开发模式，由实体（Model），视图（View），控制器（Controller）组成。

总体设计

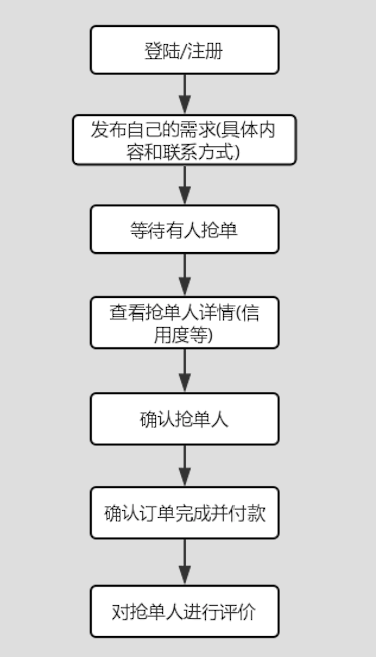
模块划分

表-1 模块划分表

| 序号 | 模块名称 | 功能说明 |
| --- | --- | --- |
|
| 1 | 地图展示首页 | 方便直接的定位自己的位置以及周围的勤人 |
| 2 | 发布页 | 发布自己的需求，寻找勤人来解决 |
| 3 | 订单页 | 可查看自己的待完成订单、已完成订单、待评价订单以及待接单订单并进行一系列操作 |
| 4 | 个人中心 | 昵称、头像、修改密码等个人信息 |
| 5 | 更多 | 清空缓存，反馈意见，关于我们，技术支持等 |
| 6 | 搜索页 | 搜索相关工作 |
| 7 | 抢单页 | 及时抢单不错过自己感兴趣的每一单 |

模块调用关系

Lazy端



Crazy端

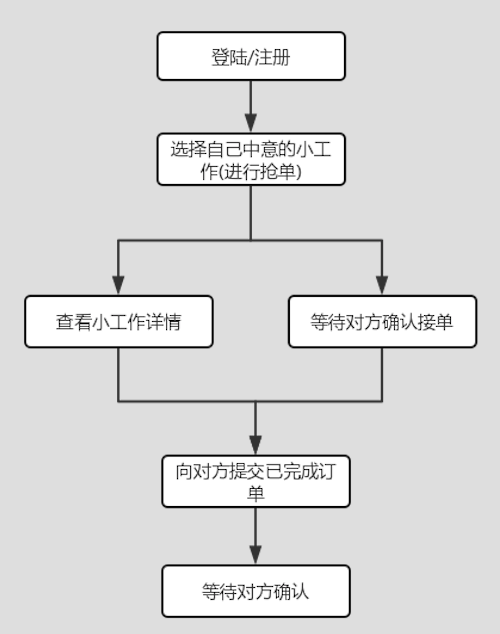


图-1 系统模块图

数据库设计

## 外部设计

### 标识符

本数据库名称为lazyorcrazy，lazyorcrazy软件名称标记。

## **支持软件**

数据库采用MySQL。

### 数据库命名规则

(1) 采用26个英文字母（**使用小写**）和0~9这十个自然数，加上下划线\_组成，共63个字符。不能出现其他字符（注释除外）。

(2) 所有的数据对象绝对不能使用汉语拼音或其缩写来命名。

### 数据库对象命名规则

(1) 数据库对象包括表、视图（查询）、存储过程（参数查询）、索引，触发器。对象 名字由前缀和实际名字组成，长度不超过30；

(2) 数据表能用一个英文单词表示，就尽量不要缩写；

(3) 全部使用小写字母。

****

### 字段命名规则

字段命名中所有单词紧靠在一起，字段所有字母均使用小写，可使用\_进行分隔。

## 数据库的逻辑结构设计

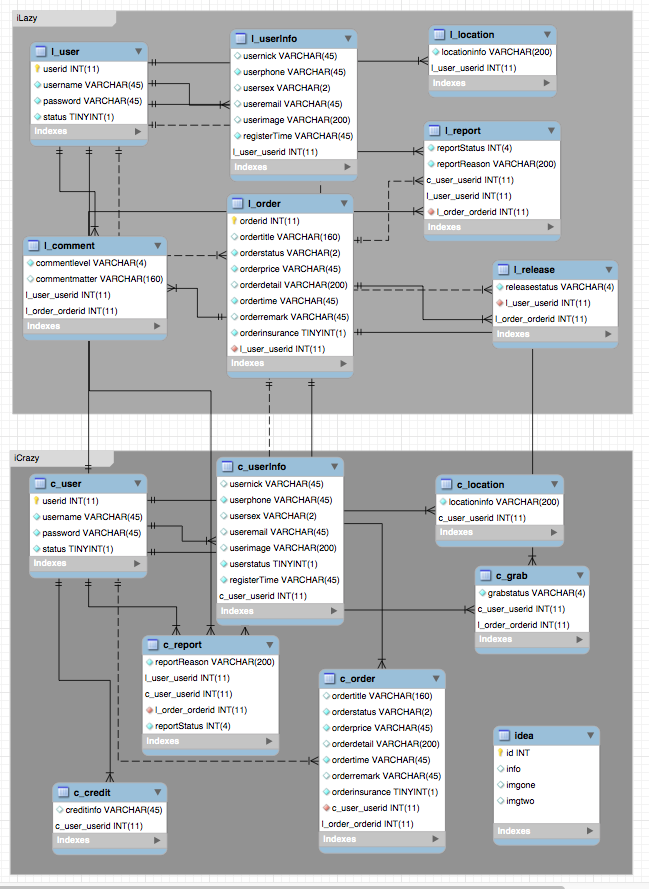


图-2 系统E-R 图

## 数据库的物理设计

(1) 主要是表结构设计。一般地，实体对应于表，实体的属性对应于表的列，实体之间的关系为表之间的约束。逻辑设计中的实体转换成物理设计中的表，但是它们并不一定是一一对应的；

(2) 对表结构进行规范化处理（第三范式）。

### 表汇总

表-3 数据表

|  |  |  |
| --- | --- | --- |
| 序号 | 名称 | 用途与约束 |
| 1 | l\_user | 懒人端用户表 |
| 2 | l\_userinfo | 懒人端用户信息表 |
| 3 | l\_release | 懒人端发布表 |
| 4 | l\_order | 懒人端订单表 |
| 5 | l\_location | 懒人端位置信息表 |
| 6 | l\_comment | 懒人端用户评价表 |
| 7 | l\_report | 懒人端用户举报表 |
| 8 | c\_user | 勤人端用户表 |
| 9 | c\_userinfo | 勤人端用户信息表 |
| 10 | c\_grab | 勤人端抢单表 |
| 11 | c\_order | 勤人端订单表 |
| 12 | [c\_credit](http://10.204.1.43:8888/phpMyAdmin/tbl_structure.php?db=fruitdb&table=t_mood&token=f36b9c682fa08b88d224446d79e784c8) | 勤人端信用表 |
| 13 | c\_location | 勤人端位置信息表 |
| 14 | c\_report | 勤人端用户举报表 |
| 15 | [idea](http://10.204.1.43:8888/phpMyAdmin/tbl_structure.php?db=fruitdb&table=t_moodimage&token=f36b9c682fa08b88d224446d79e784c8) | 用户意见反馈表 |

表-4 懒人端用户表

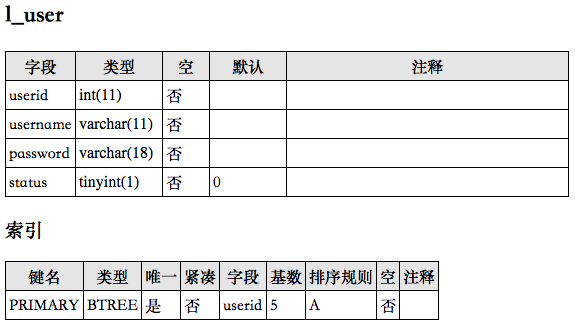


表-5 懒人端用户信息表

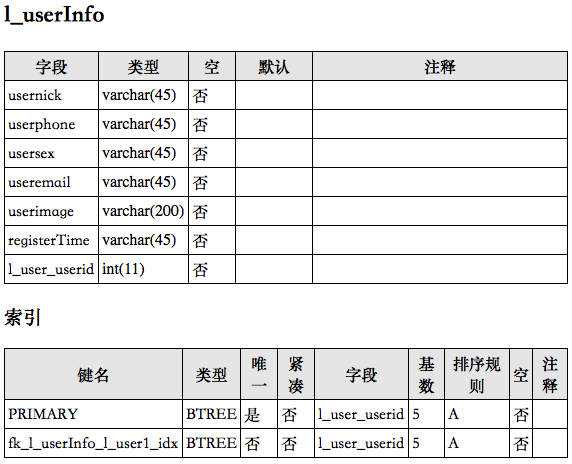


表-6 懒人端发布表

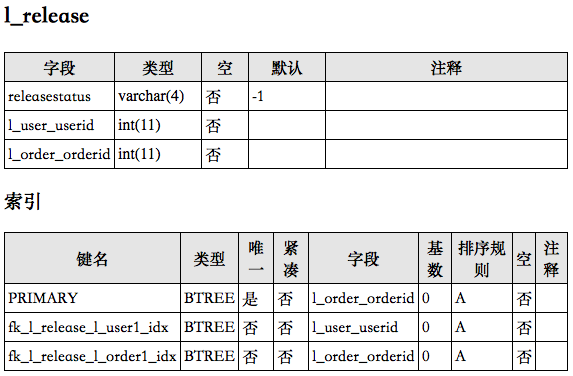


表-7 懒人端订单表

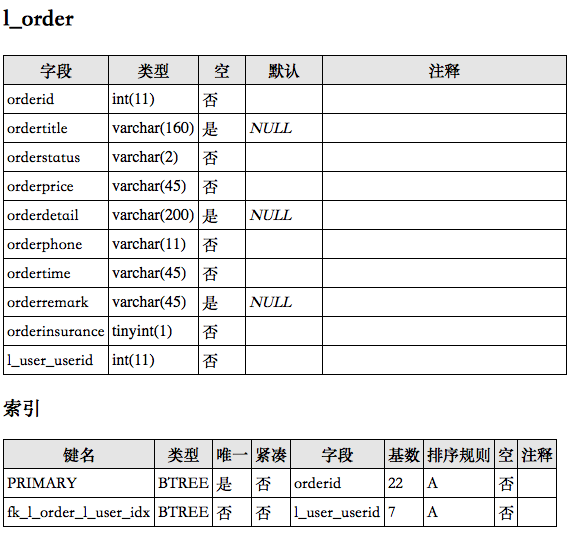


表-8 懒人端位置信息表

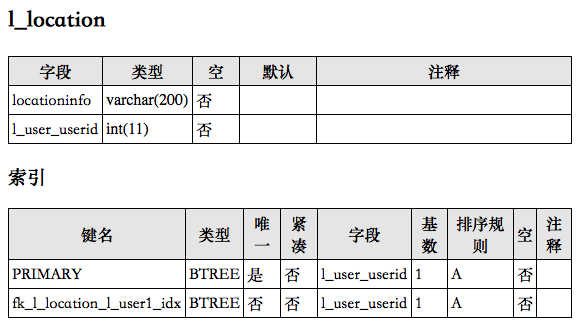


表-9 懒人端用户评价表

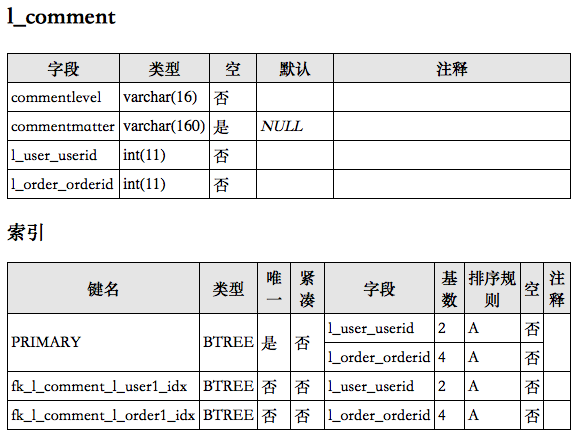


表-10 懒人端用户举报表

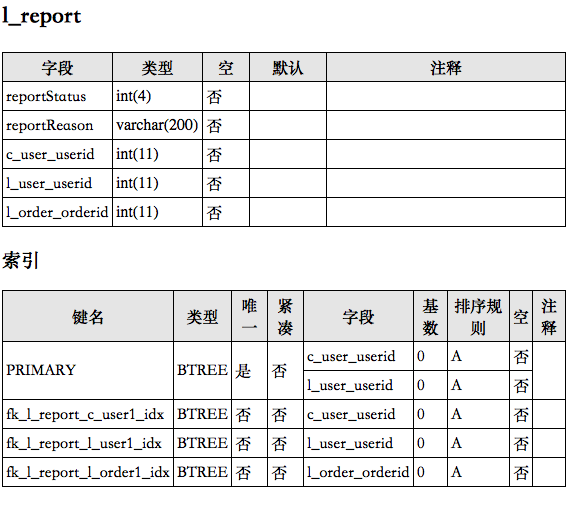


表-11 勤人端用户表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段说明 | 类型 | 长度 | NULL | DEFAULT | PK | UK | FK |
| 1 | userid | 用户表id | int | 11 | NOT |  | Y | Y |  |
| 2 | username | 用户名 | varchar | 11 | NOT |  |  |  |  |
| 3 | password | 密码 | varchar | 18 | NOT |  |  |  |  |
| 4 | status | 标识 | int | 1 | NOT | 0 |  |  |  |

表-12 勤人端用户信息表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段说明 | 类型 | 长度 | NULL | DEFAULT | PK | UK | FK |
| 1 | c\_user\_userid | 主键 | int | 11 | NOT |  | Y |  | Y |
| 2 | usernick | 昵称 | varchar | 45 | NOT |  |  |  |  |
| 3 | userphone | 联系电话 | varchar | 45 | NOT |  |  |  |  |
| 4 | useremail | 邮箱 | varchar | 45 | NOT |  |  |  |  |
| 5 | userimage | 头像地址 | varchar | 200 | NOT |  |  |  |  |
| 6 | usersex | 性别 | varchar | 2 | NOT |  |  |  |  |
| 7 | registertime | 注册时间 | varchar | 45 | NOT |  |  |  |  |
| 8 | userstatus | 标识 | int | 2 | NOT | 0 |  |  |  |

表-13 勤人端抢单表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段说明 | 类型 | 长度 | NULL | DEFAULT | PK | UK | FK |
| 1 | c\_user\_userid | 主键 | int | 11 | NOT |  | Y |  |  |
| 2 | l\_order\_orderid | 外键 | int | 11 | NOT |  |  |  | Y |
| 3 | grabstatus | 标识 | varchar | 2 | NOT | 0 |  |  |  |

表-12 勤人端订单表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段说明 | 类型 | 长度 | NULL | DEFAULT | PK | UK | FK |
| 1 | c\_user\_userid | 用户id | int | 11 | NOT |  | Y |  |  |
| 2 | l\_order\_orderid | 订单id | int | 11 | NOT |  |  |  | Y |
| 3 | ordertitle | 订单标题 | varchar | 100 | NOT |  |  |  |  |
| 4 | orderphone | 订单电话 | varchar | 11 | NOT |  |  |  |  |
| 5 | |  |  | | --- | --- | |  | orderdetail | | 订单详情 | varchar | 200 | NOT |  |  |  |  |
| 6 | orderprice | 佣金 | varchar | 45 | NOT |  |  |  |  |
| 7 | orderremark | 备注 | varchar | 45 | YES |  |  |  |  |
| 8 | ordertime | 时间 | varchar | 45 | NOT |  |  |  |  |
| 9 | orderinsurance | 保险 | int | 2 | NOT | 0 |  |  |  |
| 10 | orderstatus | 标识 | int | 2 | NOT | 0 |  |  |  |

表-14 勤人端信用表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段说明 | 类型 | 长度 | NULL | DEFAULT | PK | UK | FK |
| 1 | c\_user\_userid | 用户id | int | 11 | NOT |  | Y |  |  |
| 2 | |  |  | | --- | --- | |  | **creditinfo** | | 信用度 | varchar | 11 | YES |  |  |  |  |

表-15 勤人端位置信息表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段说明 | 类型 | 长度 | NULL | DEFAULT | PK | UK | FK |
| 1 | c\_user\_userid | 用户id | int | 11 | NOT |  | Y |  |  |
| 2 | locationinfo | 位置信息 | varchar | 11 | NOT |  |  |  |  |

表16 勤人端用户举报表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段说明 | 类型 | 长度 | NULL | DEFAULT | PK | UK | FK |
| 1 | l\_user\_userid | 懒人id | int | 11 | NOT |  | Y |  |  |
| 2 | c\_user\_userid | 勤人id | int | 11 | NOT |  |  |  | Y |
| 3 | l\_order\_orderid | 订单id | int | 11 | NOT |  |  |  | Y |
| 4 | |  |  | | --- | --- | |  | **reporteason** | | 举报内容 | varchar | 200 | NOT |  |  |  |  |
| 5 | |  |  | | --- | --- | |  | **reportstatus** | | 评论标识 | int | 2 | NOT | 0 |  |  |  |

表-17 用户意见反馈表

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 字段名 | 字段说明 | 类型 | 长度 | NULL | DEFAULT | PK | UK | FK |
| 1 | |  |  | | --- | --- | |  | **id** | | 主键 | int | 11 | NOT |  | Y |  |  |
| 2 | info | 反馈内容 | varchar | 1000 | NOT |  |  |  |  |
| 3 | imgone | 图片路径 | varchar | 200 | YES |  |  |  | Y |
| 4 | imgtwo | 图片路径 | varchar | 200 | YES |  |  |  |  |

### 

## 安全性设计

在系统中，根据用户权限获得对数据库的操作权，例如管理员可以对所有数据表进行操作，而普通用户只能对部分数据表操作。

### 防止用户直接操作数据库的方法

用户只能用账号登陆到OA系统，通过OA系统访问数据库，而没有其他途径操作数据库。

### 用户账号密码的加密方法

对用户登录密码进行加密处理，确保在任何地方都不会出现密码的明文。

## 数据库管理与维护说明

对数据库中的记录定期清理，以保证数据库的健壮性。

出错和维护设计

出错处理

出错信息

包括出错信息表、故障处理技术等。

表-29 出错信息表

|  |  |  |
| --- | --- | --- |
| **错误编号** | **出 错 信 息** | **原 因** |
| 1 | 不显示相关的数据 | 可能数据库服务器没有开启，网线没有插好 |
| 2 | 登陆失败 | 忘记了密码，没有连接数据库，网线没有插好 |
| 3 | 数据发送失败 | 服务器过忙，网线接口有断点，数据库容量超标 |

检查好物理的器件是否正常连接；数据库数据备份和删除部分数据。

扑救措施

1. 后备技术 当原始系统数据万一丢失时启用的副本的建立和启动的技术。从副本中导出原来的数据，然后再附加到数据库中，或是直接在数据库中将副本导入通过处理再次作为数据库的版本。
2. 恢复和再启动技术使系统从故障点恢复执行或使系统从头开始重新运行的方法。当系统出现故障，数据库中的数据也将更新以保存的数据，对没有进行保存的数据不保存，将初始到最后一次保存。