去年的蛋糕软件设计规格说明

Ver:1.0

**目 录**

[1 引言 3](#_Toc301301027)

[1.1 目标 3](#_Toc301301028)

[1.2 文档范围 3](#_Toc301301029)

[1.3 术语和缩略语 3](#_Toc301301030)

[1.4 参考资料 3](#_Toc301301031)

[1.5 系统目标和约束 3](#_Toc301301032)

[2 系统设计 4](#_Toc301301033)

[2.1 系统架构概述 4](#_Toc301301034)

[2.2 对象模型 4](#_Toc301301035)

[2.3 接口 4](#_Toc301301036)

[2.4 特性实现 4](#_Toc301301037)

[2.5 错误代码 9](#_Toc301301038)

[2.6 错误日志 9](#_Toc301301039)

[2.7 部署视图 9](#_Toc301301040)

[3 数据库设计 11](#_Toc301301041)

[3.1 逻辑模型 11](#_Toc301301042)

[3.2 物理模型 11](#_Toc301301043)

[4 质量及其他方面 12](#_Toc301301044)

[4.1 可维护性 12](#_Toc301301045)

[4.2 安全性 12](#_Toc301301046)

[4.3 可扩展性 12](#_Toc301301047)

[4.4 可靠性 12](#_Toc301301048)

[4.5 可用性 12](#_Toc301301049)

[4.6 性能设计 13](#_Toc301301050)

[5 附录 13](#_Toc301301051)

[5.1 附件 13](#_Toc301301052)

[5.2 修过记录 13](#_Toc301301053)

# 引言

## 目标

在完成软件开发前期的准备工作如项目需求等，结合《去年的蛋糕软件需求确认书》，项目小组(Turings小组)提出了这份软件设计说明书。

此概要设计说明书对去年的蛋糕软件进行功能分配，模块划分，程序的总体结构，输入输出和接口设计，运行设计，数据结构设计及出错设计等方面作了全面的概括性的说明，为软件详细设计奠定基础。

## 文档范围

本文档包括软件的架构设计、数据库设计、安全性、可靠性、可用性等方面的规划和设计。

## 术语和缩略语

## 参考资料

部分内容参考了《去年的蛋糕软件需求确认书》

## 系统目标和约束

软件目标：项目需要完成商家入住，订单管理，加入购物车，蛋糕搜索，下单配送等基本的买家和卖家管理。

软件的约束：因为时间有限，该软件只实现最基本的功能。

# 系统设计

## 系统架构概述

该系统的架构采用两层架构的模式：展示层（视图+业务逻辑）和数据访问层。采用面过程的设计方法。

## 对象模型

该系统采用面向对象的设计方法。

## 接口

buy();

sell();

register();

## 特性实现

1. 买家卖家注册登录
   1. 用户注册页面
   2. 注册成功，进入首页面
   3. 
   4. 
2. 卖家买家更改个人信息
   1. 点击进入个人信息
   2. 更改成功，保存退出
3. 买家蛋糕详情
   1. 点击蛋糕
   2. 查看的详细信息
4. 买家下单
5. 卖家更改蛋糕详细信息
   1. 蛋糕的增删改查
6. 客服中心

7 、蛋糕订单动态管理

8 、 根据关键信息查询特定类型的蛋糕

## 错误代码

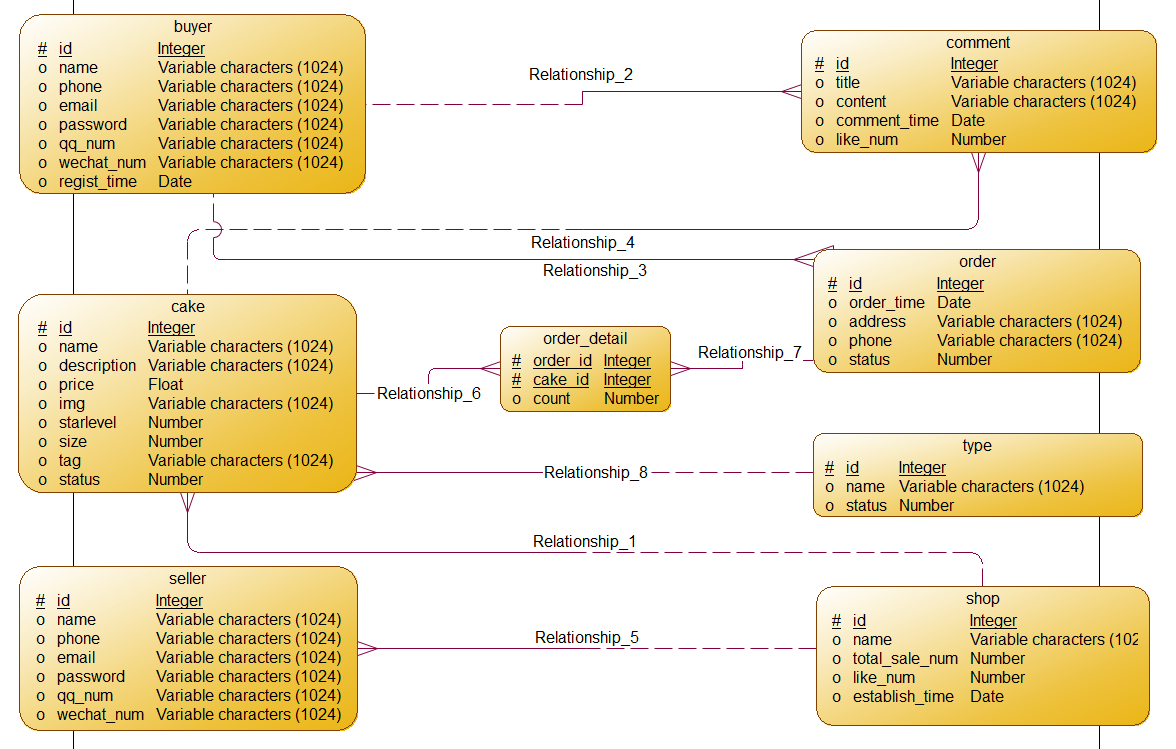
## 错误日志

## 部署视图

1. 硬件环境：
   1. 需要互联网
   2. 至少需要一台电脑和一部安卓手机。
2. 软件：
3. 需要MySQL 2015及以上和eclipse+JDK（1.8）。
4. 需要Tomcat9.0为WEB服务器。
5. 需要AndroidStudio软件。
6. 配置：
7. 打开mysql绿色版，启动服务器；
8. 下载Tomcat9.0压缩后解压缩，找到/bin目录；
9. 在AndroidStudio中添加网络权限。
10. 部署：
11. 配置eclipse连接Tomcat。
12. 配置eclipse连接MySQL，修改IP地址、数据库的用户名和密码。
13. 将安卓手机与eclipse相关联,实现前端与后台的交互。

# 数据库设计

## 逻辑模型



## 物理模型

1. 为每个表初始设置300M的存储空间，以10%的大小扩展。
2. 为上传的图片预留空间。

# 质量及其他方面

## 可维护性

1. 选择合适的高级语言，容易编程，更容易维护。
2. 通过程序注释等方式增加代码的可读性和可维护性。
3. 建立规范、完整、一致的程序文档，提高程序的可理解性
4. 将数据访问层分离，做成一个个函数，由其它层调用，以增加代码的可维护性。
5. 数据库有日志记录，系统一旦出现故障有恢复到故障之前的信息和数据的能力
6. 进行质量保证审查。质量保证审查可分为四种类型：在检查点进行检查、验收检查、周期性维护检查、对软件包检查。

## 安全性

1. 密码使用md5加密
2. 对用户的输入进行验证
3. 对用户的输入的特殊字符进行转义，防止sql注入攻击

## 可扩展性

1. 可以通过增加硬件资源的方式提高系统的响应速度。
2. 可以通过修改代码，扩充系统的功能。
3. 使用分布式消息队列降低耦合性。事件驱动架构指的是：在低耦合的模块之间传输事件消息，保持模块之间的松散耦合，通过事件消息来完成模块之间的通信。

## 可靠性

1. 软件的设计开发阶段，要进过充分讨论和评审，确保架构和编码的严谨。
2. 数据库按时备份，有日志记录
3. 通过加强对软件的测试，尽可能地解决软件中存在的问题，从而提高软件的可靠性。

## 可用性

1. 做好充分的前期工作，系统一旦投入使用，尽量减少宕机的次数和时间
2. 只提供用户需要的功能，少即是多
3. 拥有一致的界面元素，并能平衡无规则的元素。

## 性能设计

*[系统性能通常使用事务处理能力或资源利用率来度量。确定系统需求的方法：*

*识别约束：包括预算、时间、基础结构、可选的开发工具和技术；*

*确定功能特性：要符合使用场景和用例*

*确定负载：使用本系统的客户数量]*

# 附录

## 附件

墨刀原型：https://free.modao.cc/app/lrjXkDiB82M3v1EWtWJG9DnipAmU08P?simulator\_type=device&sticky

## 修过记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修改人 | 修改时间 | 修改内容 | 核准 |
| 1.0 | 刘鹏辉 | 2020-04-18 |  |  |