**毕业论文（设计）**

题 目： 餐厅点餐系统的实现

Title： Implementation of restaurant ordering system

二级学院：中兴通讯信息工程学院

专业班级：18中兴物联网班

姓 名：吕倩怡

学 号：20181201430130

指导教师：张雨沐

日 期：2022年5月

**诚 信 声 明**

本人声明：

1、本人所呈交的毕业设计（论文）是在老师指导下进行的研究工作及取得的研究成果；

2、据查证，除了文中特别加以标注和致谢的地方外，毕业设计（论文）中不包含其他人已经公开发表过的研究成果，也不包含为获得其他教育机构的学位而使用过的材料；

3、我承诺，本人提交的毕业设计（论文）中的所有内容均真实、可信。

作者签名： 日期： 年 月 日

**餐厅点餐系统的实现**

**摘要**

由于我国科技的不断发展，我国人民的生活素质也不断的在提升，越来越多的人开始被互联网世界的多姿多彩所吸引并加入。微信及各种科技硬件、软件的面世让人们可以利用互联网在虚拟网络中完成我们现实生活中的种种事件。近些年来，随着移动终端的迅速发展，对Android应用软件的需求也逐渐增大。本文首先简单介并分析了微信小程序点餐系统的研究现状和该系统开发的主要技术，分析现阶段社会对移动端应用的需求。然后对系统进行需求分析，明确系统开发的任务和目标。对系统进行整体总设计、划分功能模块。本文设计了一款基于微信小程序云开发控制台的 Android 餐厅点餐系统，这个系统可以提供简单方便的点餐流程给用户们及管理者们，可以让管理者随时随地关注和掌握餐厅的收入动态和点餐动态等等功能。这个微信小程序点餐系统的主要模块功能有：用户登录功能、菜单浏览、菜品信息、加入购物车、付费、评论等等功能、向用户提供数据统计和数据分析功能。点餐系统是基于Serverless的一站式后端云服务，涵盖了函数、数据库、存储以及CDN等服务，无需后端运维。基于云开发可以免鉴权调用微信所有开放能力。

**关键词：**微信小程序；云开发；CMS；点餐系统

**Implementation of Restaurant Ordering System**

**Abstract**

Due to the continuous development of China's science and technology, the quality of life of our people is also improving. More and more people are attracted and joined by the colorful world of the Internet. With the advent of wechat and various technology hardware and software, people can use the Internet to complete various events in our real life in the virtual network. In recent years, with the rapid development of mobile terminals, the demand for Android application software has gradually increased. Firstly, this paper briefly introduces and analyzes the research status of wechat applet ordering system and the main technology of the system development, and analyzes the current social demand for mobile applications. Then analyze the requirements of the system and clarify the tasks and objectives of system development. The overall design of the system and the division of functional modules. This paper designs an Android restaurant ordering system based on wechat applet cloud development console. This system can provide simple and convenient ordering process to users and managers, and enable managers to pay attention to and master the revenue dynamics and ordering dynamics of the restaurant anytime and anywhere. The main module functions of this wechat applet ordering system include: user login function, menu browsing, dish information, adding shopping cart, payment, comment and other functions, and providing users with data statistics and data analysis functions. The ordering system is a one-stop back-end cloud service based on serverless, which covers functions, databases, storage, CDN and other services without back-end operation and maintenance. Cloud based development can call all open capabilities of wechat without authentication.

**Key words:** WeChat applet; cloud development; CMS; ordering system

目 录

[1绪 论 1](#_Toc30149)

[1.1 课题研究背景 1](#_Toc11757)

[1.2 国内外研究现状 1](#_Toc30990)

[1.3 课题研究意义 2](#_Toc24841)

[1.4 论文主要工作和结构安排 2](#_Toc8303)

[2相关技术介绍 4](#_Toc16876)

[2.1 微信小程序 4](#_Toc11607)

[2.2 云开发 4](#_Toc28319)

[2.3 CMS内容管理系统 5](#_Toc1697)

[3需求分析 6](#_Toc14265)

[3.1 系统功能需求分析 6](#_Toc21255)

[3.2 系统性能需求分析 7](#_Toc26505)

[3.3 系统可行性分析 7](#_Toc24626)

[4系统设计 9](#_Toc22368)

[4.1 系统总体功能设计 9](#_Toc844)

[4.2 用户主要功能设计 9](#_Toc27731)

[4.2.1 登录模块设计 9](#_Toc18251)

[4.2.2 餐厅位置地图设计 10](#_Toc7922)

[4.2.3 扫码点餐模块设计 11](#_Toc7742)

[4.2.4 排号等位模块设计 11](#_Toc3521)

[4.2.5 菜品加购模块设计 12](#_Toc13933)

[4.2.6 下单模块设计 13](#_Toc22652)

[4.3 管理员主要功能设计 14](#_Toc22792)

[4.4 数据库设计 14](#_Toc12903)

[4.4.1 系统E-R图设计 14](#_Toc10165)

[4.4.2 数据库表设计 16](#_Toc19514)

[5系统实现 19](#_Toc30537)

[5.1 用户模块实现 19](#_Toc31348)

[5.1.1 登录模块实现 19](#_Toc14687)

[5.1.2 餐厅地理位置模块实现 20](#_Toc25353)

[5.1.3 扫码点餐模块实现 21](#_Toc13242)

[5.1.4 排号等位模块实现 22](#_Toc5583)

[5.1.5 菜品加购模块实现 23](#_Toc31626)

[5.1.6 下单模块实现 24](#_Toc19739)

[5.2 管理员模块实现 27](#_Toc7184)

[5.2.1 轮播图管理模块实现 27](#_Toc16485)

[5.2.2 排号管理模块实现 27](#_Toc19258)

[5.2.3 菜品管理模块实现 28](#_Toc238)

[5.2.4 订单管理模块实现 28](#_Toc25636)

[5.2.5 评论管理模块实现 29](#_Toc22721)

[6系统测试 30](#_Toc24227)

[6.1 系统测试的目的和意义 30](#_Toc9855)

[6.2 系统模板测试用例 30](#_Toc29777)

[6.2.1 用户登录模块 30](#_Toc17629)

[6.2.2 扫码点餐模块 30](#_Toc26277)

[6.2.3 购物车模块 31](#_Toc17083)

[6.2.4 下单模块 31](#_Toc14109)

[7结 束 语 32](#_Toc6455)

[7.1 系统功能总结 32](#_Toc27027)

[7.2 系统技术总结 33](#_Toc14134)

[7.3 系统展望 34](#_Toc27164)

[参考文献 35](#_Toc13864)

[致谢 36](#_Toc15160)

[附录A 37](#_Toc27167)

[附录B 38](#_Toc30226)

[附录C 39](#_Toc20337)

**1绪论**

**1.1 课题研究背景**

在当下，伴随着科技笼罩互联网的飞速发展，各种各样的信息传输速度是越来越快，人们生活节奏也受到了互联网发展的影响，随之带来了生活质量水平也要与之相匹配的问题。每天各种消息的发布与传播使得人们需要通过花费大量的时间在移动设施上进行，目前最常用的信息通讯软件APP即微信，其功能给人们的生活带了很多的方便快捷。例如通常情况下，人们就餐需要先到店内，进行点餐。近年来的疫情原因，导致实体餐饮店的生意遭受严重打击，在传统的就餐方式中，如果说互联网的发展使得实体餐饮店生意惨淡，那么疫情带来的影响可以说是灭顶之灾。依赖于互联网的发展，网络上得点餐软件也蓬勃发展了起来，不仅催生了像美团、饿了么等等这类外卖形式的点餐软件，而且人们在使用这些软件时不必出门即可享受到各式餐厅的美食，也出现了一种在餐厅内使用的在线点餐系统点餐的新颖方式。这种新的点餐方式使得用餐客户在一部智能手机上就能完成扫码、查看餐厅菜品、菜品加购、下单、支付、订单查看和订单评价等功能，这相对于传统的需要服务员配合的点餐方式显然更符合现在疫情防控的需求以及更加简单便捷。这种方式节约了用户的用餐时间，加快了餐厅的出餐效率，减少投入的人力物力，达到了提高服务效率和提升客户体验的最终目的[1]。

**1.2 国内外研究现状**

目前国内疫情下，人们的生活方式发生了很大的改变，而与人们生活密切相关的餐饮行业自然也与互联网技术进行中深度融合。在1970年就出现了餐饮电子服务管理系统，从历史发展来看，餐厅内的无限点餐系统需要服务员通过部分手持终端替代固定的计算机设备完成点餐及其他服务。而目前使用最普遍的基于互联网的无线点餐系统，其只需要用户使用智能手机就可以完成点餐、结束等服务。近年来出现的如美团、饿了么这种以外卖为主的点餐软件，还诞生了如海底捞等，在餐厅中使用手机、iPad自主点餐的新方式。其中使用手机点餐还主要分为使用特定APP点餐和扫描二维码点餐。近年来，随着微信的快速发展，基于微信平采的应用层出不穷，许多商家也将点餐系统部署到了微信平台上，顾客可以从微信直接登录到点餐系统中进行菜品浏览，下单与支付，这使得点餐变得更加方便与快捷[2]。

**1.3 课题研究意义**

在智能科技时代的飞速发展下，人们对生活效率和生活质量的追求也变得越来越高了，之前的点餐系统已经逐渐跟不上时代的步伐，人们需要更简单、更快捷的点餐程序。从而就出现的点餐的小程序。小程序实现了应用软件的无处不在和随时可用。只需一个微信一个网络，人们就可以随时随地访问各种基于微信小程序的应用，且无须下载和安装，轻松快捷。

对于互联网用户来说，只需要拥有一台智能手机和一个微信就可以进入小程序。就可以完成点餐的整个过程，无需和餐厅人员和其他客户排队等候接触几率，在当今疫情严重的情况下，这种点餐小程序可以减少疫情传染的可能性。而且小程序操作简单，容易上手，结合微信支付功能，能极大减少用户的用餐的等候时间[3]。

对于商家来说，餐厅免去了客人排队点餐付款的时间，让客人即坐即点，客户心情也会变得舒畅。提高了餐厅的翻台率，从而提高了利润。

对于开发者来说，小程序的开发成本非常低，只需要开发者掌握一定的前端知识就可以，后端基于云开发，无需开发者编写后端程序。把页面设计好，实现一定的前台逻辑即可完成小程序的开发[4]。

**1.4 论文主要工作和结构安排**

主要研究工作内容有以下几点：

（1）对于现在状况来对网络交易网站的全方面进行需求和可行性分析与研究之后，再通过整理出国内大多餐厅客人点餐的优势和不足，从而从中总结出关于微信小程序点餐的策划方案，来结合云开发技术总体进行开发运设计。

（2）在后台管理系统这一方面，使用微信小程序云开发自带的扩展CMS内容管理系统，让系统的操作变得更加便捷。

该论文分为以下七章：

第一章：绪论。主要是介绍系统的课题研究背景、国内外研究现状、课题研究意义介绍以及对论文主要工作和结构安排的简单介绍。

第二章：相关技术介绍。简单介绍微信小程序、云开发技术和CMS内容管理系统。

第三章：需求分析。主要从功能、性能和可行性等发面分析系统的需求。

第四章：系统设计。设计系统的整体架构，核心功能设计，最后设计数据库。

第五章：系统实现。对系统核心功能的细节进行一定的阐述，有代码中调用的方法等，列出实现图和核心代码图。

第六章：系统测试。测试主要功能，看结果和预期差别。从而达到完善优化系统的目的。

第七章：结束语。总结一下本论文所涉及的功能和使用技术，对本点餐小程序后续的功能拓展和重构计划做出一定的规划[5]。

**2相关技术介绍**

**2.1 微信小程序**

微信小程序相信大家都已经非常熟悉了，它有比较成熟的市场，很多商家和企业基本都拥有自己的维信小程序。微信小程序的技术栈偏向于前端，同时小程序的逻辑处理语言用的也是JavaScript，配置文件json等。如果开发者熟悉Vue框架，那么上手微信小程序就会变得非常简单，小程序的书写方式和Vue有着很大的相似之处。因此对于熟悉前端的开发者来说，掌握微信小程序的开发并没有门槛[6]。

**2.2 云开发**

以往来说，微信小程序后端的搭建都是通过Java或PHP等相关技术进行开发，而后端的学习成本很高，需要学习的技术框架多而复杂。这对只熟悉前端的开发者来说，想要开发一个完整的小程序应用，有着较高的门槛。

云开发是微信开发团队推出的一种基于云端的开发技术，相当于是为开发者搭建了后端应用，目的之一也是为了方便对后端技术不熟悉的开发者，通过编写简单的云函数，即可完成对数据库的访问。云函数又是基于NodeJS的一项功能，NodeJS是一个基于Chrome V8引擎的JavaScript运行环境，让JavaScript能够运行在服务端。开发者只需要在微信开发者工具中开通云开发服务，获取到自己的云开发环境，就可以拥有自己的云数据库。云数据库是基于MongoDB的非关系型文档数据库，配置容易，语法简单易学[7]。

图2.1云函数配置版本

**2.3 CMS内容管理系统**

图2.2CMS展示图

本系统用到的CMS系统是云开发技术自带的拓展功能，该系统功能也非常完善，无需开发者自己手动编写，不需要一行代码，只要申请开通就可以拥有自己的CMS系统。他能调用我们云数据库中的数据，进行增删改查。

对于不熟悉后端的开发者，再开发一套后台管理系统也是非常耗费时间的，CMS系统有很多的应用场景，没有开发成本，配置也及其简单。深受小程序开发者的喜爱[8]。

**3需求分析**

**3.1 系统功能需求分析**

功能需求分析是为了完善点餐小程序的功能结构，让该系统的一套点餐程序能够能够给符合用户需求，给用户完整的体验，用户功能大致有扫码点餐、餐厅菜品浏览、排号等位、菜品加购、下单、查看所有订单和订单评价等功能。管理员的需求就是结合CMS系统来管理餐厅后台数据，审核好订单信息。

系统功能需求有用餐用户和管理员，主要如下：

1.用户

用户注册登录成功后，进入点餐首页面，直接选择扫描餐桌上二维码进行点餐。如果餐厅用餐人数过多，可以选择首页的排号等位功能进行用餐排队，这和某些大型餐厅企业的类似，申请排队的人都会收到自己对应的排队号码。用户在小程序中可以查看浏览该餐厅的所有菜品及种类。用户对自己喜欢的菜品进行加购，加购的菜品系统会保存在用户的购物车中。用户在选好所有的菜品进行下单后，用户可以直接在小程序内部完成支付。支付完成之后，用户可以查看本次的订单信息和历史订单信息。用餐结束后，用户可以对订单进行评价，说明自己的用餐感受[9]。

该点餐小程序的用户用例图如图3.1所示：

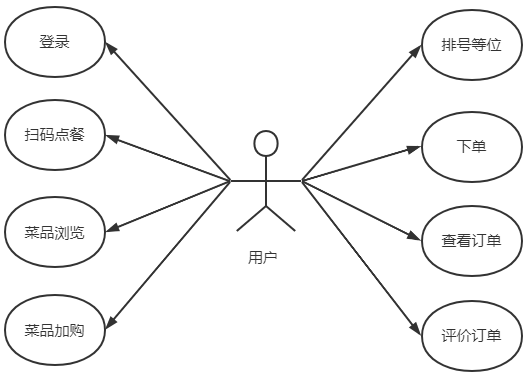


图3.1 用户用例图

2.管理员

管理员的需求基本就是管理CMS后台页面的相关信息，包括后台页面的配置信息，餐厅信息。查看确认用户订单，不能出现错误订单，影响餐厅的出餐效率和餐厅声誉。

该点餐小程序的管理员用例图如图3.2所示：

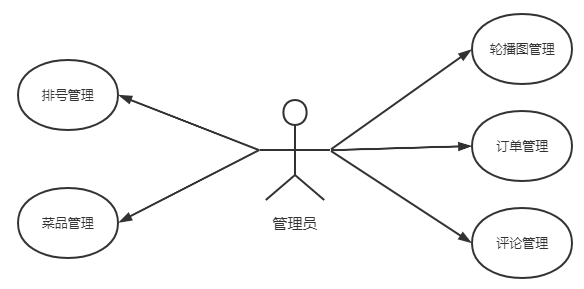


图3.2 管理员用例图

**3.2 系统性能需求分析**

1.数据安全性

数据安全性完全可以放心，云开发云数据库云资源都是微信官方的软件。微信作为腾讯公司旗下的大分支，在安全系统的保护下，我们无需担心数据会被不法人员盗取。

2.易用性

微信小程序重在灵活易用，小程序制作简单，功能不复杂，界面也是简洁明了，用户很容易进行操作。

3.扩展性

微信小程序的拓展性无需质疑，小程序的每个页面用到的资源基本都是在一个文件内部，对页面内的功能进行拓展，很容易找到对应的资源进行修改。如果要新增页面或新增功能等，系统也会自动生成开发者想要创建的页面文件夹。只要我们在设计之前考虑好了设计方案，那么操作流程也是简单快捷的。复杂的地方大概基本只有逻辑的处理了。如果要加入算法来拓展小程序功能，那么也要提前想好算法的实现思路[10]。

**3.3 系统可行性分析**

1.技术可行性分析

从技术可行性的方面来看，微信小程序的开发流程并不麻烦，开发者只需要到微信官网下载微信开发者工具，我开发用的是windows系统，所以下载windows的安装包即可，安装包安装也非常简单，基本上都是直接下一步即可，许多东西都是官方人员给配置好的，无需关心，只需要更改安装路径即可。而且小程序开发一次，即可在安卓机和苹果机同时访问。系统适配官方人员也已配置好。无需担心操作系统问题[11]。

2.经济可行性分析

从经济可行性的方面来看，该小程序的开发并不需要太多的花费，小程序的开发工具也是免费使用的，在许多配置已经存在的情况下，点餐小程序的开发周期很短就能实现。唯一需要花费的地方可能就是云开发中的云资源配置了。考虑到点餐小程序的用户使用基数，我们要保证云资源要充足，否则用户一旦多了，会造成系统卡顿问题。而云资源的价格也并不是很贵，一般只需要几十块钱就能购买到我们需要的资源量，最多不过一百多就可以，但周期是一个月的。不过总体来说是在可以接受的范围[12]。

3.操作可行性分析

从操作可行性的方面来看，该点餐小程序开发的目的就是为了简化用户在餐厅点餐的操作，不用排队等候，即扫即点，点完直接下单，下单后直接在小程序内完成支付，随后该订单信息会出现在用户手机上，用户知道自己的订单号码，坐在餐桌等候即可。在餐厅后厨也会有相应的机器来获取到用户的订单信息，厨师会根据订单内容进行菜品的制作。此过程方便快捷，省去了一定量的人力物力[13]。

4.商业可行性分析

从商业可行性的方面来看，有效形成商业闭环。让用户通过微信就可以完成菜单查看、菜品选择、下单和支付整个流程。用户有很好的体验后，就会继续使用该小程序点餐系统。从而可以将用户流量逐渐转换为餐厅利润。

**4系统设计**

**4.1 系统总体功能设计**

本餐厅点餐系统主要有用户部分的功能是模块有登录、扫码点餐、排号等位、菜品加购、下单，订单评论等功能。管理员部分主要功能是系统的轮播图管理、系统的餐厅菜品管理、系统订单管理和系统评论管理等。

系统总体的功能结构图如下图4.1所示：

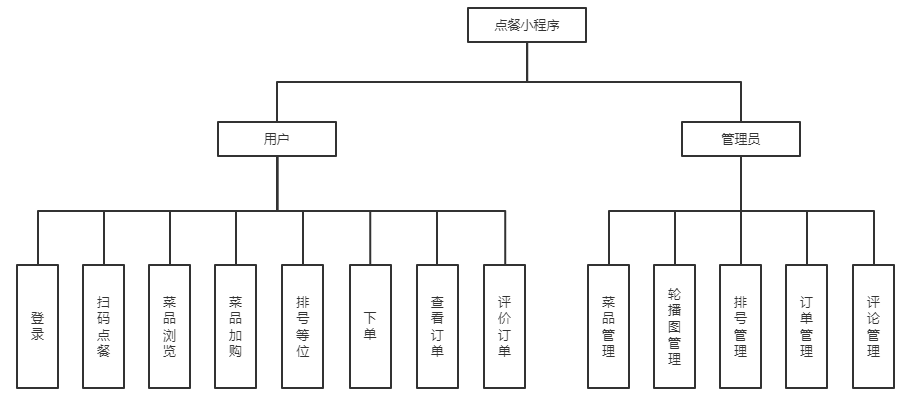


图4.1 系统功能结构图

**4.2 用户主要功能设计**

用户的最主要的功能就是点餐，这里对点餐的流程进行了设计，图4-2为用户点餐流程图。

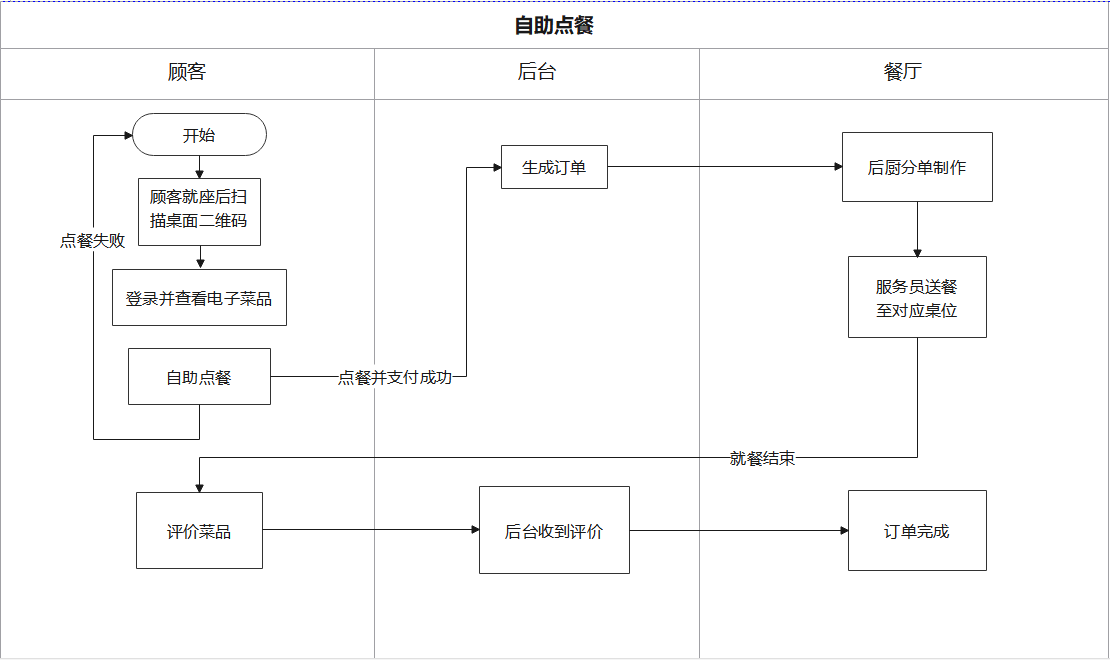


图4.2 用户点餐功能流程图

**4.2.1 登录模块设计**

用户登录的设计即采用小程序内置的登录功能。用户进入到个人界面，点击授权登录，系统会让用户确认是否同意获取用户微信相关信息。当用户同意后，登录成功，如果用户不同意，则不能完成登录。

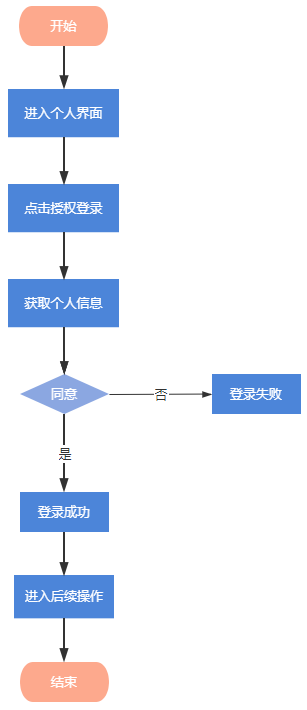


图4.3 用户登录流程图

**4.2.2 餐厅位置地图设计**

餐厅位置地图设计采用小程序自带的map组件功能。先去腾讯地图官网获取到餐厅的地理位置坐标，然后将数值记下，放到代码中。然后在wxml页面调用map组件，将位置显示到页面上。

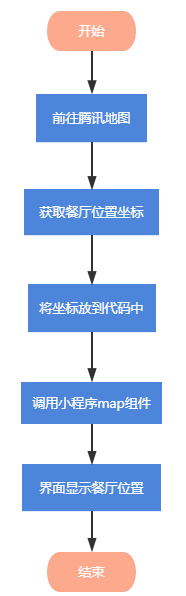


图4.4 餐厅位置地图流程图

**4.2.3 扫码点餐模块设计**

首先是将餐厅的菜品信息和二维码对应的桌号等信息制作二维码贴在对应的餐桌上，当用户点击小程序上扫码点餐按钮，系统则会识别出二维码信息。记录下桌号，将页面跳转到菜品列表页面。最后在下单时会同时将桌号信息进行显示在订单中。

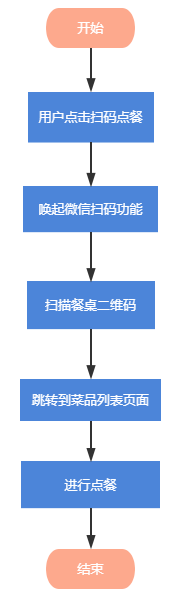


图4.5 扫码点餐流程图

**4.2.4 排号等位模块设计**

用户点击排号等位按钮，前往排号等位进行排号等位。用户先在页面上选择就餐人数，然后选择小桌就餐还是大桌就餐。如果用户选择不符合人数的餐桌，系统会有提示让用户重新选择。最后用户拿到自己的号码后进行等待。当叫到自己的号码后进行就餐。

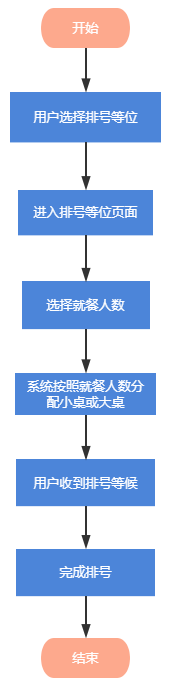


图4.6 排号等位流程图

**4.2.5 菜品加购模块设计**

用户在菜品列表页面选择菜品。第一次点击加号可以将菜品加入购物车，再次点击则将菜品数量加一，点击减号则将菜品数量进行减一。菜品列表页面底部设置一个购物车弹窗，用户选择完菜品后可以在购物车弹窗中再次对已选菜品进行操作。也可以前往购物车界面，在购物车界面，也会显示用户选择的菜品。在这里也可以对菜品进行操作。操作完成后进行下单。

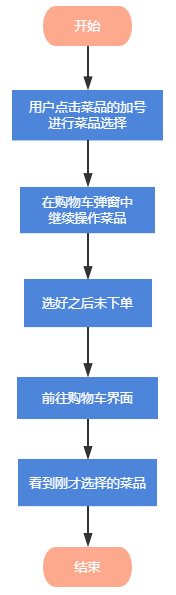


图4.7 菜品加购流程图

**4.2.6 下单模块设计**

用户在选完菜品后点击下单按钮即可进行下单操作。系统会记录用户选择的菜品和桌号信息。前往下单界面，在下单界面，显示出用户选择的菜品信息和所对应的桌号信息。用户还可以在这里选择就餐人数和添加备注。

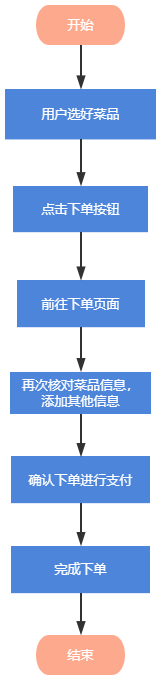


图4.8 下单流程图

**4.3 管理员主要功能设计**

管理员功能设计主要就是依赖于云开发自带的CMS内容管理系统，首先在云开发的拓展功能里面找到CMS，然后将它开通。之后云开发自动调用我们的云数据库，把数据库中的数据表数据进行封装，之后拿到系统给的CMS链接地址，在网页打开该链接，进入到我们自己的CMS管理后台页面。

后台管理页面的设计，总体是一个三栏式布局的页面，顶部是管理后台的标题信息，左侧是一些菜单列表，有餐厅的具体内容，项目配置等的展示。右侧是内容主区域，主要展示所选菜单的内容，选择内容集合里的子菜单，可以对餐厅的信息进行展示。选择设置菜单，我们可以对后台标题进行修改，对项目进行简单的介绍[14]。

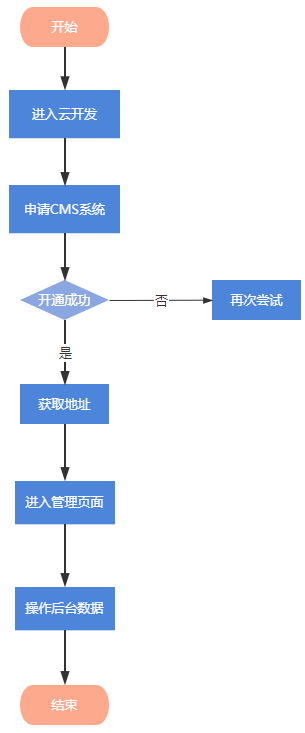


图4.9 管理员功能流程图

**4.4 数据库设计**

**4.4.1 系统E-R图设计**

数据库的设计按照该点餐小程序的功能模块来设计，数据库实体与实体之间的关系E-R图如下图4.10所示：

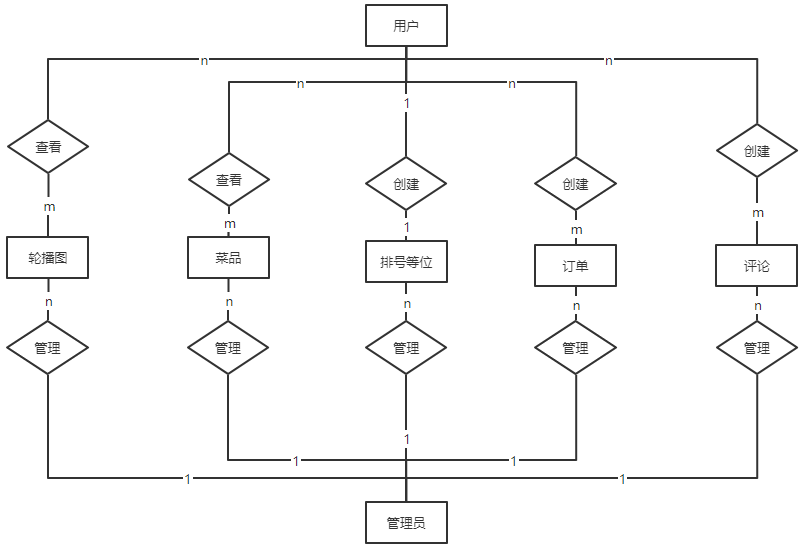


图4.10 系统E-R图

用户：用户编号、用户名、角色、创建时间、UUID。

管理员：用户编号、用户名、密码、角色、创建时间、UUID。

轮播图：轮播编号、图片路径、创建时间、更新时间。

排号：排号编号，用户名，小桌排号，大桌排号，小桌当前就餐号，大桌当前就餐号，创建时间，更新时间。

菜品：菜品编号、菜品名、菜品类目、菜品价格、菜品图片、菜品销量、菜品状态、创建时间、更新时间。

订单：订单编号、下单时间、用户open\_id、桌号、人数、备注、订单列表、订单状态、数量、总价。

评论：评论编号、用户名、用户头像、评论内容、创建时间和更新时间。

**4.4.2 数据库表设计**

本系统的设计主要主要是存储一些餐厅用到的信息，即展示到小程序中的信息，管理员可以对该数据库中的信息进行操作修改。数据库选择的是云数据库(基于MongoDB)进行数据表的建立和管理，便于数据的存储和管理。本系统有用户表（user）、管理员表（admin）、轮播图表（lunbotu）、排号表（paihao）、菜品表（food）、订单表（order）、评论表（comment）这七张表。

1.用户表user，是用来描述菜品信息的表。用户表主要包括用户编号、用户名、角色和创建时间等属性。其中用户编号为此用户实体表的主键。用户信息表的具体设计如表4.1所示：

表4.1 用户表(user)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 描述 | 字段名 | 数据类型 | 长度 | 是否为主键 |
| 1 | 用户编号 | \_id | int | 32 | Y |
| 2 | 用户名 | username | string | 32 | Y |
| 3 | 角色 | roles | string | 64 | N |
| 4 | 创建时间 | \_createTime | number | 128 | N |
| 5 | UUID | uuid | string | 128 | N |

2.管理员表admin，是用来描述管理员信息的表。管理员表主要包括用户编号、用户名、密码、角色和创建时间等属性。其中用户编号为此管理员实体表的主键，管理员信息表的具体设计如表4.2所示：

表4.2 管理员表(admin)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 描述 | 字段名 | 数据类型 | 长度 | 是否为主键 |
| 1 | 用户编号 | \_id | int | 32 | Y |
| 2 | 用户名 | username | string | 32 | N |
| 3 | 密码 | \_password | string | 32 | N |
| 4 | 角色 | roles | string | 64 | N |
| 5 | 创建时间 | \_createTime | number | 128 | N |
| 6 | UUID | uuid | string | 128 | N |

3.轮播图表lunbotu，是用来描述用管理员管理的轮播图消息，用于显示在用户页面的轮播图。轮播图表主要包括了轮播图编号、图片路径、描述和创建时间等属性。其中轮播编号为改实体表的主键，轮播图表的具体设计如表4.3所示：

表4.3 轮播图表(lunbotu)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 描述 | 字段名 | 数据类型 | 长度 | 是否为主键 |
| 1 | 轮播图编号 | \_id | string | 32 | Y |
| 2 | 图片路径 | picUrl | string | 128 | N |
| 3 | 创建时间 | \_createTime | number | 64 | N |
| 4 | 更新时间 | \_updateTime | number | 64 | N |

4.排号表paihao，是用来描述排号详细信息的表。排号表主要包括排号编号，用户名，小桌排号，大桌排号，小桌当前就餐号，大桌当前就餐号，创建时间，更新时间等属性。其中排号编号为排号表实体表的主键。排号表的具体设计如表4.4所示：

表4.4 排号表(paihao)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 描述 | 字段名 | 数据类型 | 长度 | 是否为主键 |
| 1 | 排号编号 | \_id | string | 64 | Y |
| 2 | 用户名 | username | string | 32 | N |
| 3 | 小桌排号 | smallTable | number | 32 | N |
| 4 | 小桌排号 | bigTable | number | 32 | N |
| 5 | 小桌当前就餐号 | smallTableCurrent | number | 32 | N |
| 6 | 大桌当前就餐号 | bigTableCurrent | number | 32 | N |
| 8 | 创建时间 | \_createTime | number | 128 | N |
| 9 | 更新时间 | \_updateTime | number | 128 | N |

5.菜品表food，是用来描述菜品详细信息的表。菜品表主要包括菜品编号、菜品名、菜品类目、菜品价格、菜品图片、菜品销量、菜品状态等属性。其中菜品编号为菜品表实体表的主键。菜品表的具体设计如表4.5所示：

表4.5 菜品表(food)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 描述 | 字段名 | 数据类型 | 长度 | 是否为主键 |
| 1 | 菜品编号 | \_id | string | 64 | Y |
| 2 | 菜品名 | name | string | 64 | N |
| 3 | 菜品类目 | fenlei | string | 32 | N |
| 4 | 菜品价格 | price | number | 32 | N |
| 5 | 菜品图片 | icon | string | 64 | N |
| 6 | 菜品销量 | sell | number | 32 | N |
| 7 | 菜品状态 | status | string | 32 | N |
| 8 | 创建时间 | \_createTime | number | 128 | N |
| 9 | 更新时间 | \_updateTime | number | 128 | N |

6.订单表order，是用来描述订单信息的表。订单表主要包括了订单编号、下单时间、桌号、菜品编号、订单状态、数量、总价、备注和评论内容等属性。其中订单编号为该订单实体表的主键，订单表的具体设计如表4.6所示：

表4.6 订单表(order)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 描述 | 字段名 | 数据类型 | 长度 | 是否为主键 |
| 1 | 订单编号 | \_id | string | 32 | Y |
| 2 | 下单时间 | create\_time | number | 64 | N |
| 3 | 用户open\_id | \_openid | string | 32 | N |
| 4 | 桌号 | address | string | 32 | N |
| 5 | 人数 | renshu | number | 32 | N |
| 6 | 备注 | beizhu | string | 32 | N |
| 7 | 订单列表 | orderList | array | 32 | N |
| 5 | 订单状态 | staus | number | 32 | N |
| 6 | 数量 | count | int | 32 | N |
| 7 | 总价 | totalPrice | string | 32 | N |

7.评论表comment，是用来描述用管理员管理的订单评论信息，用于收集用户对用餐体验的描述。评论表主要包括了评论编号、用户名、用户头像、评论内容、创建时间和更新时间等属性。其中评论编号为改实体表的主键，轮播图表的具体设计如表4.7所示：

表4.7 评论表(comment)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 描述 | 字段名 | 数据类型 | 长度 | 是否为主键 |
| 1 | 评论编号 | \_id | string | 32 | Y |
| 2 | 用户名 | username | string | 128 | N |
| 4 | 评论内容 | content | string | 128 | N |
| 5 | 创建时间 | \_createTime | number | 64 | N |
| 6 | 更新时间 | \_updateTime | number | 64 | N |

**5系统实现**

**5.1 用户模块实现**

该餐厅点餐系统中的用户模块主要包含的功能模块有登录、信息管理、订单管理、评价管理。用户通过微信授权登录后，打开系统首先就会看见菜单，选择菜品后可以查看详细介绍和用户评价，之后加入购物车，提交订单信息给管理员审核，管理员审核通过后进行下单，然后等待上菜，用餐结束后可对消费菜品进行评价，另外在个人信息中可查看历史定单与评价信息。

**5.1.1 登录模块实现**

登录管理模块使用微信授权登录管理实现，在底部导航栏我的界面中，最上方有一个绿色的登录按钮，当用户点击该按钮后，会有请求授权的弹窗，待用户点击允许后，系统调用的是小程序内置的wx.getUserProfile方法，该方法会将用户的微信相关信息如用户昵称、头像、openid等返回到后台。若在一段时间内有过登录操作，系统会将该用户信息在storage中，当用户再次登录时候无需再次授权。具体登录界面如图5.1所示，核心代码图如图5.2所示。



图5.1 登录界面图



图5.2 登录核心代码

**5.1.2 餐厅地理位置模块实现**

餐厅的地图设计用的是微信小程序内置的腾讯地图，首先要明确一个事情，任何位置都有它的经纬度 ，所以要先获取到我们想定位位置的经纬度。那么就是到腾讯地图上查询餐厅在现实位置中的经纬度。获取到餐厅位置的经纬度坐标值longitude和latitude，将这两个值放到address文件夹中的address.js的data中，再调用小程序内置的map组件，将longitude和latitude值进行绑定，那么地图上就会自动定位到我们选择的位置上了。另外还将地址的详细信息在地图下方用文字进行描述。餐厅地理位置图如图5.3所示。核心代码图如图5.4所示。



图5.3 餐厅地理位置图

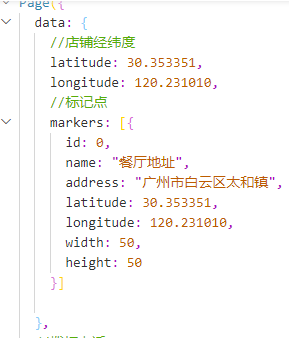


图5.4 餐厅地理位置核心代码图

**5.1.3 扫码点餐模块实现**

用户使用微信的扫码功能扫描餐桌上二维码时，系统调用home文件夹下的home.js文件中的scanCode方法，在方法内部调用微信内置的扫码功能方法，wx.scanCode，在该方法中有两个重要参数onlyFromCamera，这里我们设置为false，让用户只能通过手机相机现场扫描餐桌的二维码。当用户成功扫码后，调用wx.scanCode方法的成功回调函数success，将页面跳转到菜单页面，系统调用云函数getFoodList,查询云数据库food的菜品，将菜品状态status为上架的商品进行获取。通过js逻辑渲染到菜单页面。供用户挑选菜品。扫码点餐图如图5.5所示。核心代码图如图5.6所示。



图5.5 扫码点餐图



图5.6 扫码点餐核心代码图

**5.1.4 排号等位模块实现**

用户点击首页的排号等位选项，进入到paihao.wxml页面，用户点击选择人数，来确认当前需要就餐的人数，系统通过用户选择的人数，通过调用bindPickerChange方法，来确定用户就餐人数。系统调用paihao方法，来确定用户的排号信息。当用户就餐人数为1-4人时，系统为用户分配小桌；当用户就餐人数为5-10人时，系统为用户分配大桌。最后系统调用addPaiHaoInfo，在方法内部调用云函数paihao，将用户的排号信息保存到云数据库paihao中。排号等位图如图5.7所示。核心代码图如图5.8所示。



图5.7 排号等位图



图5.8 排号等位核心代码图

**5.1.5 菜品加购模块实现**

用户在菜单页面选择菜品，系统会调用setStorageSync方法将菜品暂时缓存到购物车中，用户可以在菜单页面唤起购物车弹窗对已选菜品进行编辑。系统在food.js中调用addCount方法和reduceCount方法对菜品数量进行编辑。用户点击下方的购物车导航后，系统调用getStorageSync方法，将缓存中的购物车信息取出，渲染到页面中进行展示。购物车图如图5.9所示。核心代码图如图5.10所示。



图5.9 购物车图



图5.10 购物车核心代码图

**5.1.6 下单模块实现**

用户在菜单页面选择完菜品后，点击选好了按钮。系统会将用户选择的菜品信息保存在storage中，并跳转到pay文件夹下的pay.wxml页面，在pay.js中把storage缓存的菜品信息取出来显示在点餐明细中。这里还可以看到桌号信息，用户也可以输入用餐人数和备注信息。待用户完全确认后，点击下方的绿色按钮，进行付款，系统会调出选择支付的弹窗，用户可以选择支付方式进行支付。最后系统js中的submitOrder方法，在方法内部调用云函数db.collection("order").add方法，将用户的下单信息保存到云数据库order中。用户下单图如图5.11所示。核心代码图如图5.12所示。



图5.11 用户下单图



图5.12 用户下单核心代码图

**5.2 管理员模块实现**

**5.2.1 轮播图管理模块实现**

管理员进入CMS后台，点击左侧的内容集合中的轮播图菜单。即可通过内置的云函数get\_carouselImg, 在函数内部通过编写数据库get脚本，访问云数据库的lunbotu表，将轮播图的信息渲染到页面中的表格中。管理员可以点击新建按钮调用云函数的insert\_ carouselImg方法，在函数内部通过编写数据库insert脚本，对轮播图中的信息进行新增，点击右侧的编辑按钮，调用云函数的update\_ carouselImg，在函数内部通过编写数据库update脚本，对已有的轮播图信息进行编辑，点击删除按钮，调用云函数的delete\_ carouselImg，在函数内部通过编写数据库delete脚本，对轮播图进行删除。也可以点击导入数据，将外边的轮播图数据进行直接导入，点击导出数据，将已有的轮播图信息以json文件形式进行保存。轮播图管理图如图5.13所示。

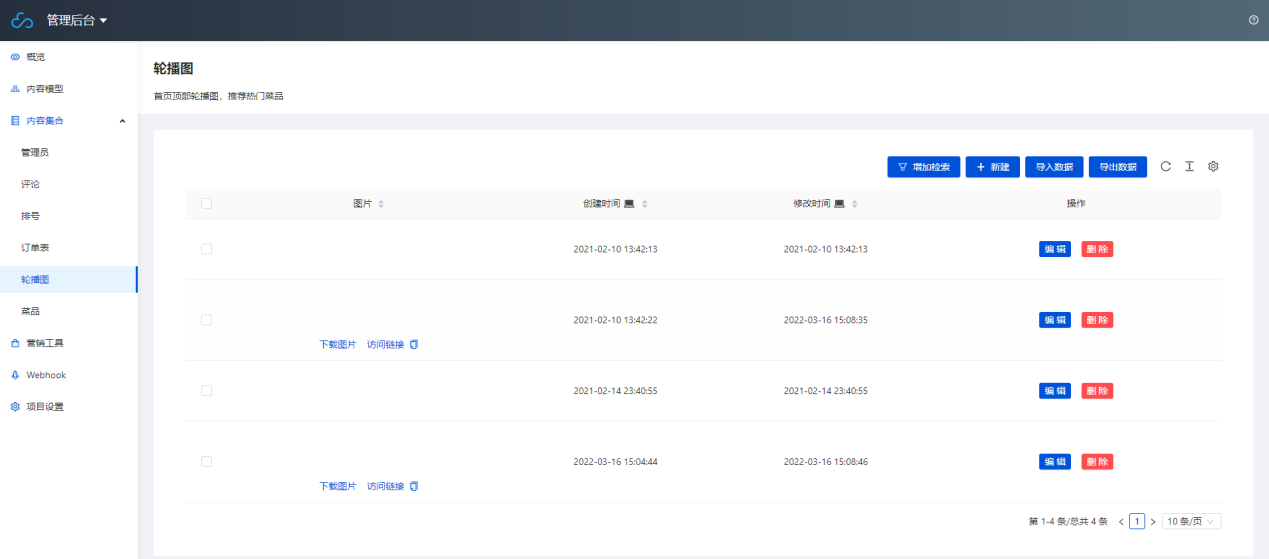


图5.13 轮播图管理图

**5.2.2 排号管理模块实现**

管理员进入CMS后台，点击左侧的内容集合中的排号菜单。即可通过内置的云函数get\_ rowNumber, 在函数内部通过编写数据库get脚本，访问云数据库的rowNumber表，将排号的信息渲染到页面中的表格中。管理员可以点击新建按钮调用云函数的insert\_rowNumber方法，在函数内部通过编写数据库insert脚本，对排号中的信息进行新增，点击右侧的编辑按钮，调用云函数的update\_rowNumber，在函数内部通过编写数据库update脚本，对已有的排号信息进行编辑，点击删除按钮，调用云函数的delete\_rowNumber，在函数内部通过编写数据库delete脚本，对排号进行删除。也可以点击导入数据，将外边的排号数据进行直接导入，点击导出数据，将已有的排号信息以json文件形式进行保存。排号管理图如图5.14所示。



图5.14 排号管理图

**5.2.3 菜品管理模块实现**

管理员进入CMS后台，点击左侧的内容集合中的菜品菜单。即可通过内置的云函数get\_food, 在函数内部通过编写数据库get脚本，访问云数据库的food表，将菜品的信息渲染到页面中的表格中。管理员可以点击新建按钮调用云函数的insert\_food方法，在函数内部通过编写数据库insert脚本，对菜品中的信息进行新增，点击右侧的编辑按钮，调用云函数的update\_food，在函数内部通过编写数据库update脚本，对已有的菜品信息进行编辑，点击删除按钮，调用云函数的delete\_food，在函数内部通过编写数据库delete脚本，对菜品进行删除。也可以点击导入数据，将外边的菜品数据进行直接导入，点击导出数据，将已有的菜品信息以json文件形式进行保存。菜品管理图如图5.15所示。

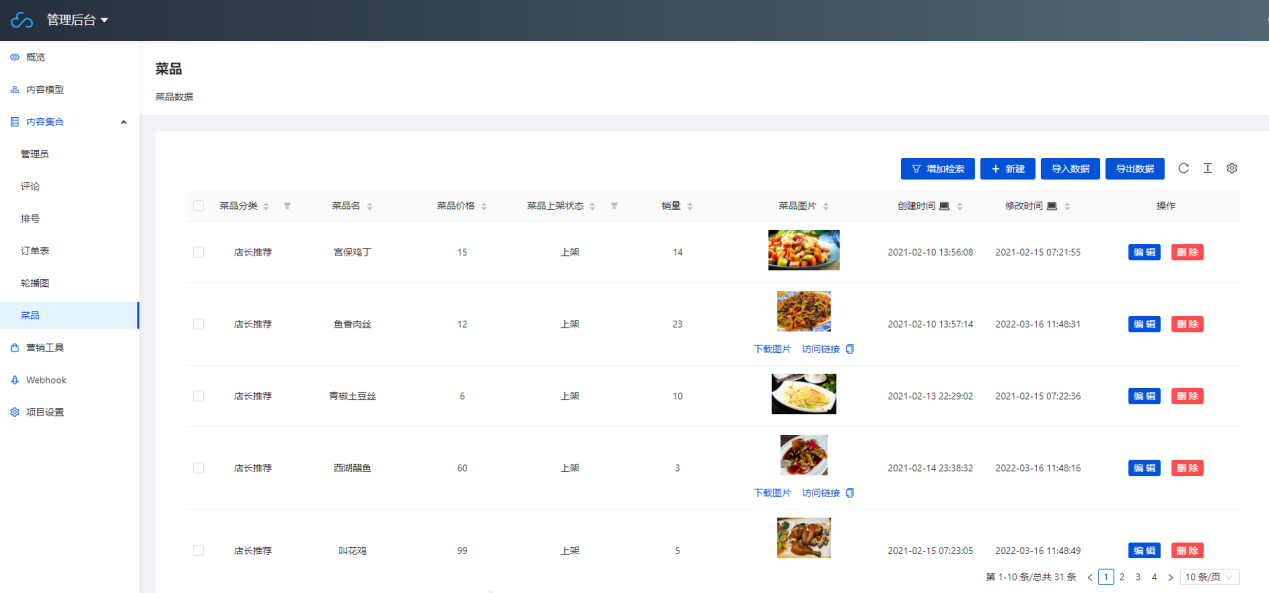


图5.15 菜品管理图

**5.2.4 订单管理模块实现**

管理员进入CMS后台，点击左侧的内容集合中的订单表菜单。即可通过内置的云函数get\_order,在函数内部通过编写数据库get脚本，访问云数据库的order表，将订单的信息渲染到页面中的表格中。管理员可以点击新建按钮调用云函数的insert\_order方法，在函数内部通过编写数据库insert脚本，对订单中的信息进行新增，点击右侧的编辑按钮，调用云函数的update\_order，在函数内部通过编写数据库update脚本，对已有的订单信息进行编辑，点击删除按钮，调用云函数的delete\_order，在函数内部通过编写数据库delete脚本，对订单进行删除。也可以点击导入数据，将外边的订单数据进行直接导入，点击导出数据，将已有的订单信息以json文件形式进行保存。订单管理图如图5.16所示。



图5.16 订单管理图

**5.2.5 评论管理模块实现**

管理员进入CMS后台，点击左侧的内容集合中的评论菜单。即可通过内置的云函数get\_comment，在函数内部通过编写数据库get脚本，访问云数据库的comment表，将评论的信息渲染到页面中的表格中。管理员可以点击新建按钮调用云函数的insert\_comment方法，在函数内部通过编写数据库insert脚本，对订单评论中的信息进行新增，点击右侧的编辑按钮，调用云函数的update\_comment，在函数内部通过编写数据库update脚本，对已有的评论信息进行编辑，点击删除按钮，调用云函数的delete\_comment，在函数内部通过编写数据库delete脚本，对评论进行删除。也可以点击导入数据，将外边的评论数据进行直接导入，点击导出数据，将已有的评论信息以json文件形式进行保存。评论管理图如图5.17所示。

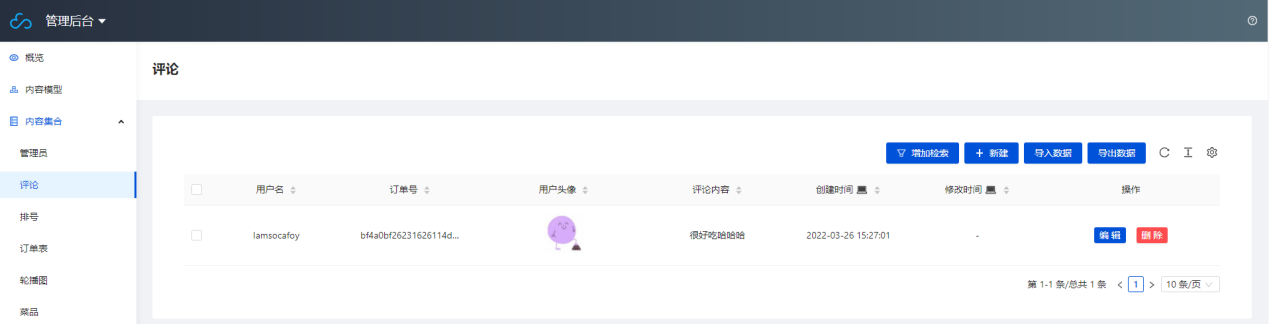


图5.17 评论管理图

**6系统测试**

**6.1 系统测试的目的和意义**

系统测试的目的非常明显，就是让我们辛辛苦苦制作的系统尽可能的减少bug，优化用户的使用体验，软件的最终形态是要展现给用户的，让用户进行使用。如果用户使用我们的软件总是出现意外的系统漏洞，影响了他们的使用体验，就会造成很多不必要的纠纷，让餐厅流失一定数量的客户，降低餐厅收益，总之会有很多不好的影响。我们应当在不同的场景下进行测试，这样能保证测试的质量，能查到更多潜在的系统错误，同时也要兼顾到用户在不同操作系统下使用点餐小程序是否会有系统漏洞。或许有的功能在安卓机正常运行，但是在苹果机上就会有意外发生。总之，系统测试工作不得马虎，要细致和全面[15]。

**6.2 系统模板测试用例**

**6.2.1 用户登录模块**

功能特性：通过调用微信接口，获取用户的微信昵称和头像等信息。

测试目的：查看是否在条件满足下获取到用户的这些信息，便于在后面用户操作时候携带上用户的这些信息。

测试结果：结果和预期基本相同，当条件满足时候，能够获取到用户的微信昵称和头像等信息，条件不满足时候无法获取到这些信息，测试用例如下表6.1所示。

表6.1 用户登录模块表

|  |  |  |
| --- | --- | --- |
| 输入 | 应输出 | 实际输出 |
| 授权管理√ 系统权限√ | 正常显示用户信息 | 正常显示用户信息 |
| 授权管理√ 系统权限× | 无法获取用户信息 | 无法获取用户信息 |
| 授权管理× 系统权限√ | 无法获取用户信息 | 无法获取用户信息 |
| 授权管理× 系统权限× | 无法获取用户信息 | 无法获取用户信息 |

**6.2.2 扫码点餐模块**

功能特性：用户做到餐厅位置上之后，扫描桌上二维码进行点餐，该功能只能进行现场扫码，不能在手机图库中扫码。

测试目的：让后台正确收到用户的餐桌信息和其他信息等。

测试结果：结果和预期基本一致，正常情况下用户扫码都会跳转到菜单页面，后台正确记录用户餐桌信息，如果出现网络卡顿或扫码不正确，后台不会收到餐桌信息。所以餐桌信息必须要正确，这影响到用户的用餐体验和后厨出餐的顺序。测试用例如下表6.2所示。

表6.2 扫码点餐模块表

|  |  |  |
| --- | --- | --- |
| 输入 | 应输出 | 实际输出 |
| 1号桌二维码 | 进入点餐桌号1 | 进入点餐桌号1 |
| 2号桌二维码 | 进入点餐桌号2 | 进入点餐桌号2 |
| 扫描二维码名片 | 返回主界面 | 返回主界面 |
| 扫描连接二维码 | 返回主界面 | 返回主界面 |

**6.2.3 购物车模块**

功能特性：将用户的菜品选择放入小程序自带的缓存。

测试目的：菜品购物车中显示的菜品种类和数量都与用户选择的菜品和数量一一对应，没有出错。能正确将用户的菜品选择保存到缓存中。

测试结果：结果和预期基本相同，用户对菜品购物车的操作基本不会出现数据不对应的错误，测试用例如下表6.3所示。

表6.3 购物车模块表

|  |  |  |
| --- | --- | --- |
| 输入 | 应输出 | 实际输出 |
| 点击+加入菜品到购物车 | 该菜品数量+1 | 该菜品数量+1 |
| 点击—减少菜品数量 | 1变0 且菜品消失  或数量-1 | 数量为1的菜品消失  数量大于1的菜品数量-1 |
| 点击清空购物车 | 购物车变空 | 购物车变空 |
| 点击遮罩层 | 收起购物车界面 | 收起购物车界面 |

**6.2.4 下单模块**

功能特性：记录用户的订单内容信息，保存云数据库，并传给后厨进行菜品制作。

测试目的：在前台有新用户使用小程序扫码点餐下单后，系统数据库能正确保存到该订单信息，后厨也能正确收到该订单信息。

测试结果：结果与预期基本相同，在用户下单后，数据库能保存到用户的订单信息，后厨也能收到对应的订单信息，测试用例如下表6.4所示。

表6.4下单模块表

|  |  |  |
| --- | --- | --- |
| 输入 | 应输出 | 实际输出 |
| 用户下单付款完成后 | 后端显示新订单信息 | 后端显示新订单信息 |
| 点击显示新订单 | 跳转到订单显示界面 | 跳转到订单显示界面 |

**7结 束 语**

**7.1 系统功能总结**

本毕业设计设计开发的餐厅点餐系统，主要功能为用户端的用户点餐行为和管理员端对订单和菜品管理功能的实现。

用户端实现的主要功能有，餐厅的菜品浏览、用户扫码点餐下单、用户菜品加购、用户排号等位、用户订单查看等。这些功能都依赖于云开发技术，有的是调用云函数进行云数据库的操作，有的直接在页面的js文件唤起云数据库，通过async/await异步函数直接进行操作。很好的将数据在用户端和云端进行交互。成功实现的预期的所有功能。

管理员端实现的主要功能有轮播图信息管理、餐厅菜品信息管理、订单信息管理、评价信息管理和用户订单评价管理，这些都依托于云开发的拓展功能CMS管理系统。这套拿来即用的系统，结合云数据库可以让我们很好的管理该点餐小程序的大部分信息。从而及时将信息进行优化变更。

**7.2 系统技术总结**

小程序页面使用的前端技术是wxml类html，wxss类css，js，json等。这些都是小程序自带的文件。当在pages文件夹下创建一个页面文件夹时候，这些会自动生成。设备适配问题用小程序自带的css单位rpx解决。

扫码点餐用的是小程序原生的ScanCode来调起相机，同时将onlyFromCamera属性设置为true，让用户只能进行现场扫码。使用makePhoneCall功能来进行电话预约。

**7.3 系统展望**

未来的发展方向，我打算将点餐系统的排队功能进行拓展，现在的排号等位功能还过于简单，在排队人数少的情况下可以使用，但一旦人数增加到了一定数量后，将会面临排队的桌号重复的问题，如何使用合适的算法将该功能进行优化，是当下要解决的问题之一。其二将小程序的内容布局变得多样。将点餐小程序增加一定活动、优惠、特色菜品，推出更多的菜系种类，让小程序的内容看起来更加丰富。最后就是小程序目前的页面看起来还不是非常美观，后续会利用网上好看的小程序ui框架，对点餐小程序的界面进行重构。给用户更好的体验感[。

**参考文献**

1. 申燕萍,何梦磊.基于微信小程序的点餐系统[J].电脑知识与技术:学术版,2018.
2. 沈东旭,白建,姚嘉明.基于小程序在市场交易中的应用[J].河北农机2020(02):74.
3. 蒙海莹.在线评测微信小程序的设计与实现[J].湖北农机化2020(01):139-140.
4. 袁堂青,亓婧.基于微信小程序的开发与研究[J].网络安全技术与应用,2020(04):66-67.
5. 2019年微信小程序数据[J].中国科技信息,2020(08):8-11.
6. 邓斌权,李剑波,瞿先超.基于云开发和微信小程序的垃圾分类系统实现[J].电脑知识与技术,2020,16(07):82-84.
7. 余丽娜.微信小程序订餐系统设计与实现[J].信息通信,2020(02):131-132.
8. 张利香,甘发旺,薄建国.基于微信小程序与云开发的成绩查询系统实现[J].内蒙古科技与经济,2019(13):59-60.
9. 广兰路地铁.微信小程序云开发HTTPAPI调用[J].TIPS,2020.

[10]韩玉坤.微信小程序对APP市场扰动的博弈论分析[J].商业经济,2020(01):70-73.

[11]周亚楠,张佳仪,任雨杰,史文娟.基于大数据理论指导下的高校食堂点餐微信小程序功能设计[J].电脑知识与技术,2019,15(34):55-59+68.

[12]陈思,冷雪.微信小程序开发方式对比[J].电子制作,2020(02):52-53+22.

[13]吴明桦,李杰.微信小程序的优势分析及其在企业中的应用[J].电子技术与软件工程,2020(15):45-46.

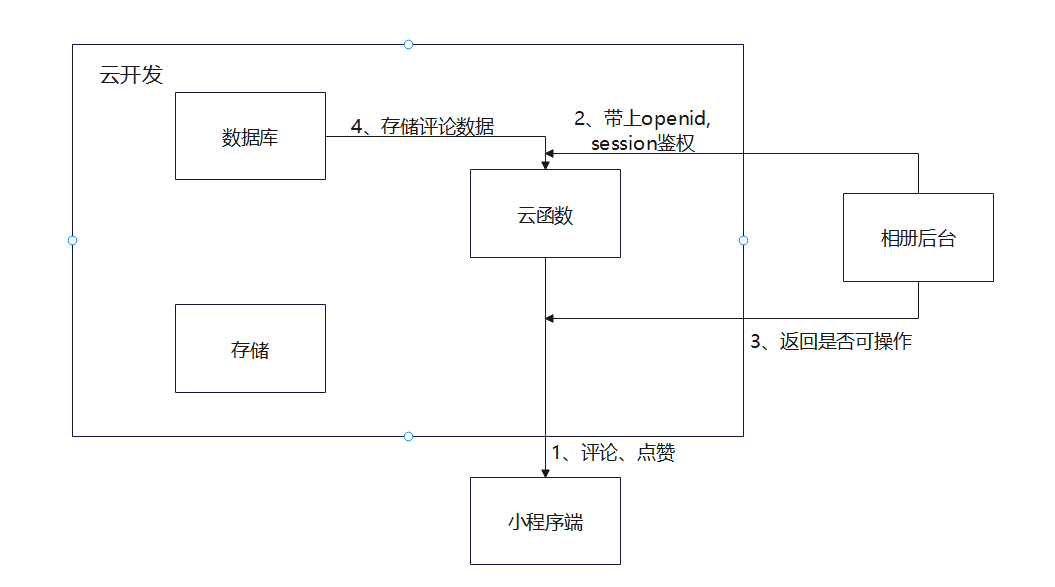
[14]陈建潮.基于“微信小程序”新形势下的互联网+档案服务平台研究[J].网络安全技术与应用,2019(12):163-164.

[15]谢文兰,何小平.基于Bmob后端云的微信小程序[1]内蒙古科技与经济,2020(04):66-68.

# **致谢**

本论文（小程序设计）是在张雨沐老师的细心和专业指导下按时完成的。此时此刻已临近毕业即将离开传授了我四年知识的学校，随着论文的逐步完成我深切领略到教师职业的深刻含义，在设计实现的过程中，张雨沐老师以他渊博的学识、严谨的治学态度、务实的工作作风和敏捷的思维，深深的影响着我，时刻鞭策着我向优秀靠近，感谢他为我们营造了浓郁的学术氛围，使我们攻克设计过程中遇到的种种难题，最终顺利完成设计，没有张雨沐老师的悉心指导和辛勤付出就没有这篇论文的产生。最后，向各位不辞辛劳审阅本论文的老师表示衷心的感谢！

# 附录A



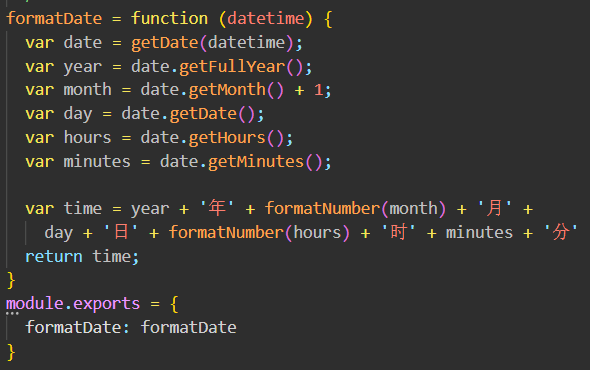
图A点餐小程序评论逻辑

附录B

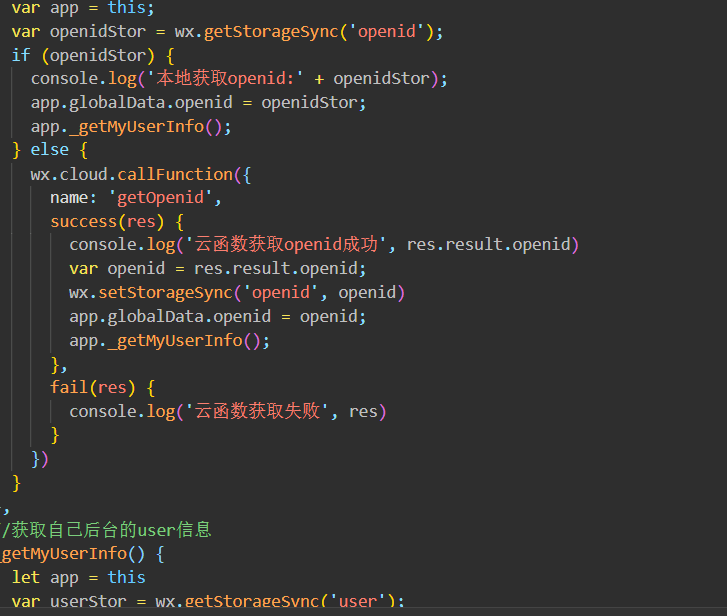


图B小程序首页设计

附录C



图C.1时间配置主函数



图C.2云开发初始化主函数