**饭店点餐微信小程序**

**系统分析与设计**

**项目组成员: 15331373 石赛军**

**15331271 申朋超**

**15331337 许博文**

**15331303 王宇翔**

**指导教师： 王青**

**第一章 需求分析**

**1.1、问题概述：**

随着网络的发展，传统的点菜模式逐渐让人觉得低效、繁碎，用电脑自动化管理餐厅点菜成将为一种不可逆转的趋势。传统点菜方式弊端很多，菜单不直观、缺少足够的服务员导致客户等待时间过长、人工失误等等。

总的来说，本项目是广泛利用信息技术，使点菜实现信息化，同时点菜系统的信息化也是以信息技术应用为基础的管理改造过程。点餐信息化实施的目的在于更好的将餐馆内部的思想进行共享，从而能够更好的提升餐馆的管理水平和管理能力。

**1.2、必要性和重要性**

1） 人们对点菜速度的要求 随着生活水平的提高，人们在餐厅就餐是对于速度的要求也是越来越高。点菜系统的电子化和信息化是大势所趋。

2） 从餐厅经营的角度看，点菜实行信息化有非常重要的意义 点菜实行电子信息化对于企业来说，一方面可以提高顾客的点菜速度，服务质量方面可以提高，另一个方面来讲，对于企业日后的发展都有十分重要的意义，有利于企业的管理和协调工作。

3） 点餐信息系统的建立能提高企业的管理水平和效率 从餐饮企业的管理来看，点餐信息化是企业提高管理效率、保证决策正确有效、促进企业发展的重要因素。而餐饮企业信息化的表现之一就点餐系统信息化。

**1.3、系统使用角色划分**

管理员、顾客

**1.4、系统设计板块划分**

主要分为四个区域

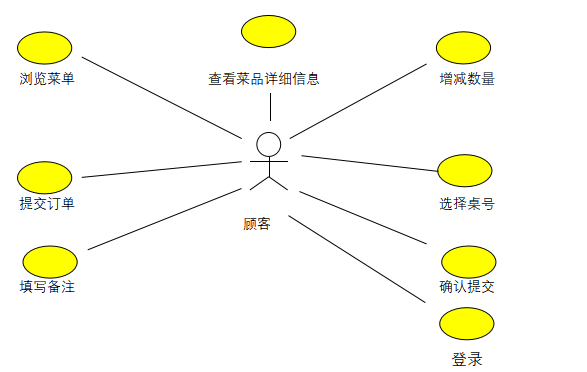
1. 宣传区（宣传特色菜品）
2. 点菜区（客人点选菜品）
3. 提交订单区（计算金额，提交订单）
4. 确认信息区（添加备注，填写桌位）

**1.5、用例析取**

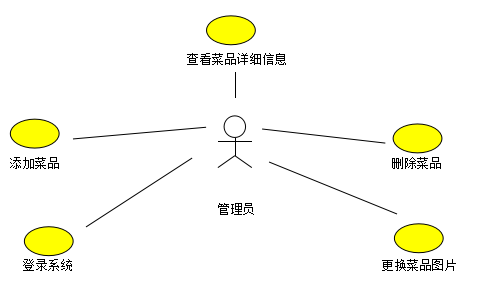
顾户：网上点餐、填写备注、提交订单等。

管理员：添加新菜品，更改菜品价格，更换菜品图片等。

**1.5.1、点餐系统用例图**



**1.5.2、管理员管理系统用例图**



**1.5.3、用例规约**

1.5.3.1、顾客点餐

1）简要说明

本用例允许顾客浏览菜单，查看菜品详细信息，增加或减少每种菜品的数量，提交订单，选择桌号，可选填备注，确定提交。

2）参与者

顾客用户

3）事件流

Ⅰ.基本事件流

用例开始于顾客使用微信扫描二维码，登录点餐小程序。

A.小程序将菜单以平铺的方式显示在屏幕上，现实的内容包括图片、价格、名字和已点数量。

B.顾客可以点击图片查看菜品的详细信息，再详细信息框中可以点击加入购物车。

C.顾客可以点击菜品下面的加减号来修改数量。

D.顾客点击“提交订单”，进入“确认菜品”界面，在此界面显示已点菜品及数量，可以选择桌号，填写备注，确认提交，弹出框显示“提交成功”。

Ⅱ.后备事件流

当未点任何菜品点击“确认提交”时，弹框提醒“没有选择商品”。

4）特殊需求

使用较新版本的微信，网络信号良好。

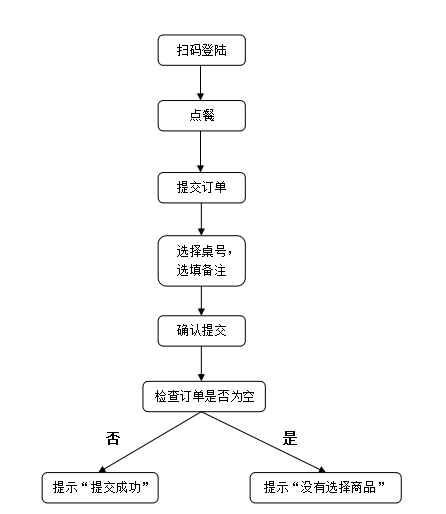
5）前置条件

本用例开始前顾客已使用微信扫码登录小程序。

6）后置条件

如果用例成功，该订单将被餐厅管理员接受，并通知厨房准备食物。

1. 活动图



1.5.3.2、管理员管理

1）简要说明

本用例允许管理员查看菜品详细信息，添加或删除菜品，更换菜品图片。

2）参与者

管理员

3）事件流

用例开始于管理员使用微信扫描二维码，登录点餐小程序。

A.小程序将菜单以平铺的方式显示在屏幕上

B.管理员可以点击图片查看菜品的详细信息

C.管理员可以点击菜品上面的图片以更换

D.管理员可以添加或删除菜品

4）特殊需求

使用较新版本的微信，网络信号良好。

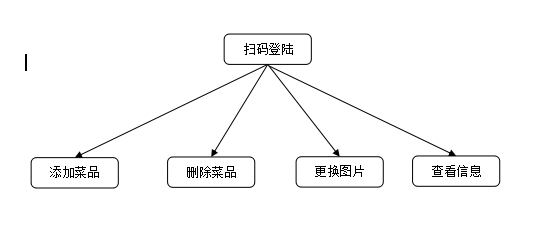
5）前置条件

本用例开始前管理员已使用微信扫码登录小程序。

6）后置条件

使用管理员账号登录

7）活动图



**1.6、补充规约**

1）可靠性

系统运行快速，不卡顿，稳定可靠且高效。本系统可以保证7\*24小时连续运行，一周维护时间不会超过2小时。

2）可用性

本系统目标群体为餐厅顾客，操作简便，界面简洁，让点餐的信息录入变得直接明了。实现信息录入、浏览、删除、修改，用户对点餐订单的提交。

3）高性能

本系统能满足餐厅满客情况下所有顾客同时使用，页面反应速度快，操作流畅。即时操作即时生效，用户获得操作反馈速度快。

4）扩展性

在结构设计上有扩展的可能，以便于未来对此系统的扩展维护升级等等。

**1.7、术语表**

|  |  |
| --- | --- |
| 术语 | 含义 |
| 顾客用户 | 所有在餐厅中扫码登录的顾客 |
| 管理员 | 餐厅负责管理点餐系统的工作人员 |
| 桌号 | 顾客所在桌子左上角的号码 |
| 备注 | 填写对菜的咸度、甜度、辣度、酱料、香料方面等的特殊要求 |

**第二章 架构设计**

**小程序页面设计基本上也是遵循 MVC 结构进行构建。**

VIEW视图层：

本层是页面显示部分主要包含

首页：

1. 推荐展示
2. 食品列表
3. 提交订单
4. 弹出面板
5. 购物车
6. 菜品标签
7. 排序标签

订单页面：

1. 桌号
2. 备注
3. 价格
4. 提交修改按钮

MODEL数据访问层：此部分用于处理数据逻辑。

本程序的模型层包含：

1. 用户页面的数据模型
2. 用户订单数据模型
3. 相关食品数据模型
4. 用户操作的数据模型。

CONTROL控制层：

控制器主要包含：

1. 用户点击按钮或图片页面跳转的控制逻辑
2. 用户将菜品放入或取出购物车、选取菜品分类和排序的控制逻辑
3. 用户对图片的拖拽控制以及选取键入订单信息控制逻辑等

**页面由4个文件构成**

1）js：页面逻辑，相当于控制层（C）；也包括部分的数据（M）

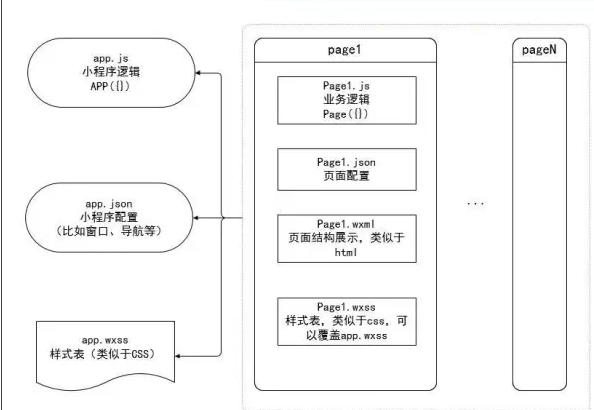
2）wxml：页面结构展示，相当于视图层（V）

3）wxss：页面样式表，纯前端，用于辅助wxml展示

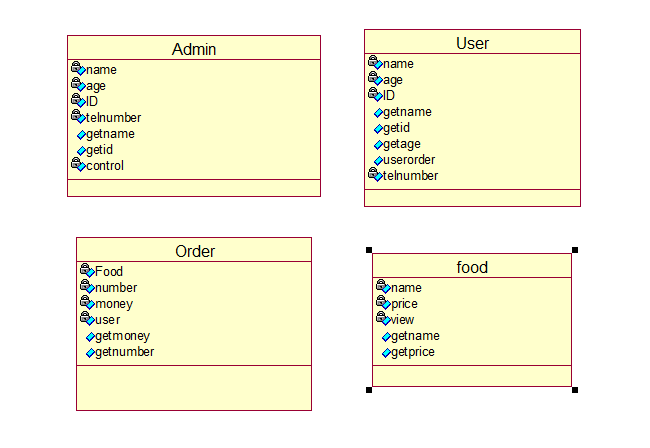
4）json：页面配置，配置一些页面展示的数据，充当部分的模型（M）

小程序架构如下：





**关键抽象**：通过数据库储存数据我们可以得到关键抽象：



**第三章 用例分析**

**3.1查看菜单详细信息用例分析**

在查看菜单详细信息的用例中，顾客用户通过菜单类发出查看菜单详细信息的请求，详细信息控制类收到请求后，从菜单数据类中读取相应的数据后返回给菜单类。查看菜单详细信息的用例分析如图3-1所示：

CustomerUser ReadDetailInf ReadDetailInfCtrl DetailInf

图3-1 查看菜单详细信息用例分析图

经过分析，查看菜单详细信息的时序图如图3-2所示：

:User :ReadDetailInf :ReadDetailCtrl :DetailInf

imageClick()

queryData()

setData()

图3-2 查看菜单详细信息用例时序图

**3.2点餐用例分析**

在点餐用例中，顾客用户首先通过微信扫码进入点餐小程序，顾客进入系统后可以 进行的操作：增加菜品数量、减少菜品数量、提交订单、选择桌号、填写备注和确认订单。点餐的用例分析如图3-3所示:

AddNumCtrl

SubNumCtrl

CustomerUser StartOrder SubmitOrderCtrl OrderForm

SelectTableNumCtrl

FillRemarkCtrl

ConfirmOrderCtrl

图3-3 点餐用例分析

经过分析，点餐的时序图如图3-4所示：

:CustomerUser :StartOrder :AddNumCtrl :SubNumCtrl :SubmitOrderCtrl :SelectTableNumCtrl :FillRemarkCtrl: ConfirmOrder:CtrlOrderForm

foodList()

addClick()

reduceClick()

sublimitClick()

tables()

remark()

okClick()

bindPickerChange()

图3-4 点餐时序图 **3.3 菜单管理用例分析**

在菜单管理用例中，管理员用户可以添加、删除、修改菜品，菜单管理的用例分析如图3-5所示：

AddMenuCtrl

AdminUser ManageMenu ReduceMenuCtrl MenuForm

ModifyMenuCtrl

:AdminUser :ManageMenu :AddMenuCtrl :ReduceMenuCtrl :ModifyMenuCtrl :MenuForm

manage

addMenu()

reduceMenu()

modifyMenu() check（）

图3-4 点餐时序图

**3.4 餐厅点餐系统合并类的析取**

ReadDetailInf ReadDetailInfCtrl DetailInf

AddNumCtrl

SubNumCtrl

CustomerUser StartOrder SubmitOrderCtrl OrderForm

SelectTableNumCtrl

FillRemarkCtrl

ConfirmOrderCtrl4

AddMenuCtrl

AdminUser ManageMenu ReduceMenuCtrl MenuForm

ModifyMenuCtrl

**3.5分析机制**

|  |  |
| --- | --- |
| 分析类 | 分析机制 |
| ReadDetailInf | 持久性、安全性 |
| StartOrder | 持久性、安全性 |
| ManageMenu | 持久性、安全性 |

**第四章 类与子系统设计**

**4.1确定类的设计**

以下是系统的总体类设计图：

ReadDetailInf ReadDetailInfCtrl DetailInf

AddNumCtrl

SubNumCtrl

CustomerUser StartOrder SubmitOrderCtrl OrderForm

SelectTableNumCtrl

FillRemarkCtrl

ConfirmOrderCtrl4

AddMenuCtrl

AdminUser ManageMenu ReduceMenuCtrl MenuForm

ModifyMenuCtrl

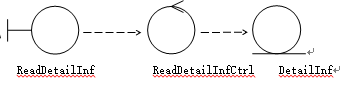
图4-1 总体类设计

## 4.2 对系统进行子系统设计

子系统设计是将某些类封装在一个子系统中，子系统可以提供一个或者多个接口，其他类或子系统可以直接通过接口获得子系统的功能，而不需要与子系统内部的类发生联系。子系统作为一个功能单元而存在于系统中。经过分析，点菜系统的子系统设计如图所示，本系统中共有三个子系统，其中ReadSubsystem子系统提供查询菜品详情功能，OrderSubsystem子系统提供点菜功能，包括菜品的添加删除，确认订单，计算金额等，MenuManageSubsystem子系统提供菜单的管理功能，包括增加删除修改。

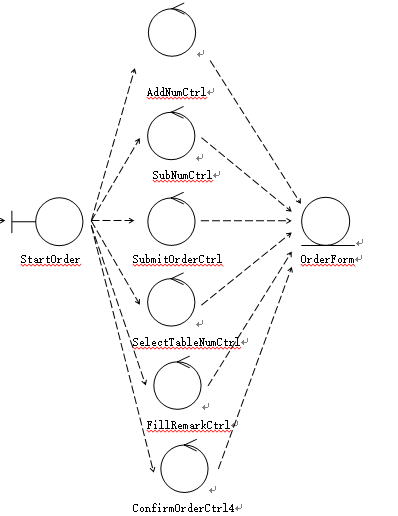
《Subsystem》

ReadSubsystem



查询子系统设计

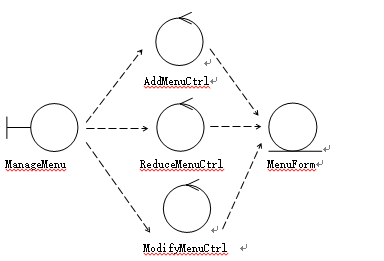
<<subsystem>>

OrderSubsystem

点菜子系统设计

<<subsystem>>

MenuManageSubsystem



管理菜单子系统设计

**4.3子系统及其接口设计**

该系统的子系统设计如上图，用户通过选择确认桌号，能够加入系统之中。

各个子系统通过边界类（接口）与其他子系统进行数据的交换，通过接口将不同的子系统的逻辑分开，数据的传输也均通过接口来传递，这样能够实现系统间的低耦合度。在调用每个子系统相应功能的时候，仅能通过该系统所暴露的接口。

而对于数据、界面、控制层则采用MVC的框架进行分层。

OrderSubsystem

User

ReadSubsystem

MenuManageSubsystem

Admin

**5 描述系统运行时架构**

**5.1分析本系统并发需求**

按照前面的分析，本系统主要分为三个模块：人员管理，菜单管理和查看信息。如图5.1:

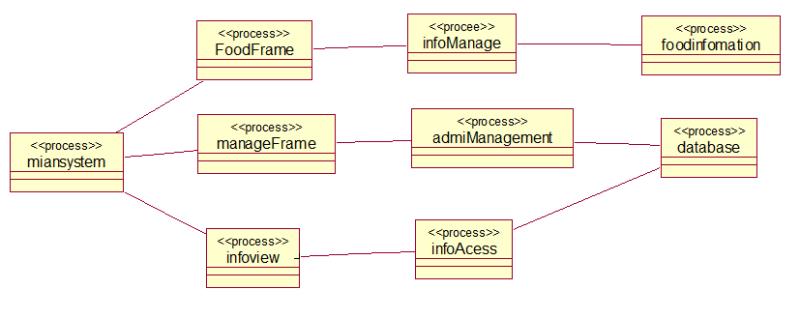


图5.1 本系统进程

**5.2识别出相应的进程和线程**

三个模块并不需要并发操作，都只需要单线程操作

**5.3描述相关的并发进程和线程**

5.3.1人员管理并发进程和线程描述

如图5.2所示

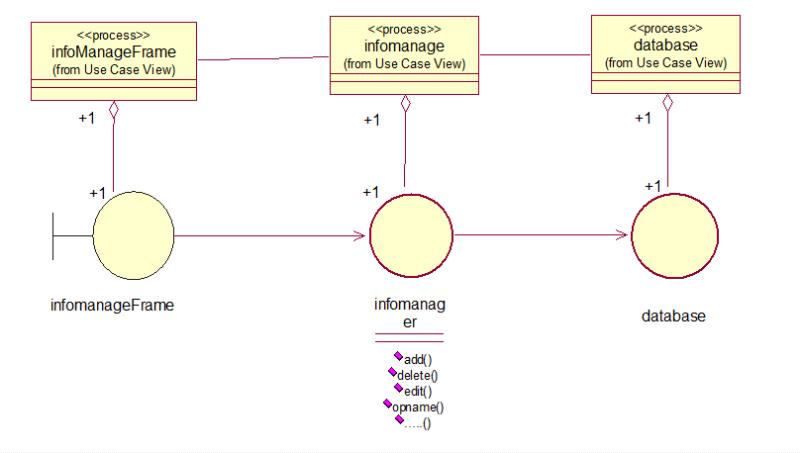


图5.2 人员管理运行时架构

5.3.2菜单管理进程和线程描述

如图5.3所示

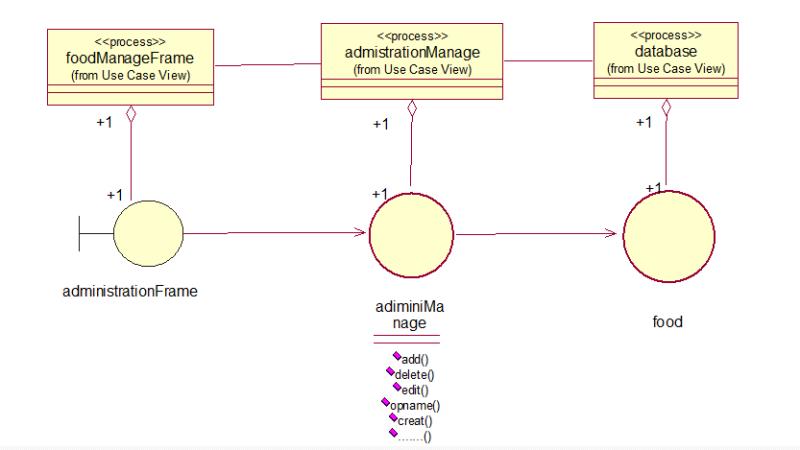


图5.3 菜单管理运行时架构

5.3.3信息查看进程和线程描述

如图5.4所示

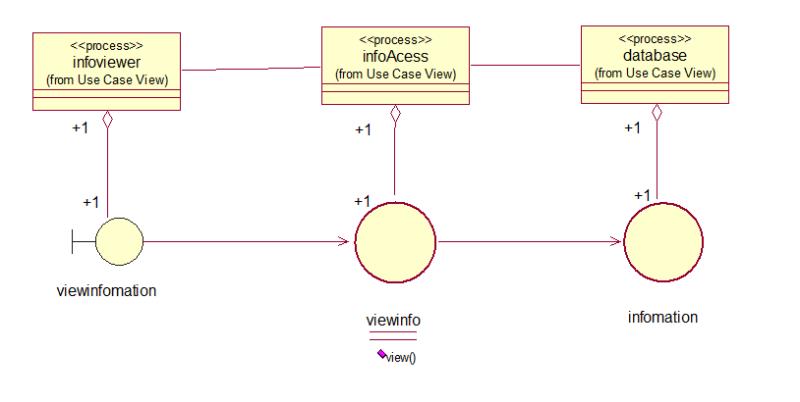


图5.4 信息查看运行时架构