



湖南工业大学科技学院
COLLEGE OF SCIENCE AND TECHNOLOGY HNUT

课程设计说明书

课程名称: Python 程序设计课程设计

设计题目: 餐馆收银系统

专 业: 计算机科学与技术 班级: 计科 1901

学生姓名: 沈俊杰 学 号: 19412030129

指导教师: 刘 强

湖南工业大学科技学院教务部 制

2022 年 5 月 27 日

目 录

1 引言	1
1.1 系统开发背景	1
1.2 研究意义	1
1.3 国内外研究现状	1
1.4 关键技术分析	2
1.4.1 SQLite5	2
1.4.2 PyQt5、PyQt5-tools	2
1.5 系统的特色和创新之处	3
1.6 本章小结	3
2 需求分析	1
2.1 可行性分析	1
2.1.1 系统实施的影响分析	1
2.1.2 技术可行性	1
2.1.3 经济可行性	1
2.1.4 操作可行性	1
2.1.5 其他因素分析	1
2.2 总体架构	2
2.3 功能需求分析	2
2.3.1 需求描述	2
2.3.2 数据流图	3
2.3.3 数据字典	4
2.4 系统的非功能性需求	5
2.4.1 性能、容量	5
2.4.2 可靠性	5
2.4.3 可维护性	5
2.4.4 安全	5
2.5 本章小结	5
3 系统设计	6
3.1 系统设计	6
3.1.1 功能模块图	6
3.1.2 系统功能分析	6
3.1.3 系统用例图等	7
3.2 数据库设计	7
3.2.1 局部概念模式设计	7
3.2.2 全局概念模式设计	8
3.2.3 数据库设计	9
3.3 本章小结	10
4 系统设计与实现	1
4.1 系统相关功能的实现	1
4.1.1 注册会员	1

4.1.2	换绑手机	2
4.1.3	添加菜品	5
4.1.4	删除菜品	6
4.1.5	修改菜价	8
4.1.6	账单结算-加菜	11
4.1.7	打印小票	13
4.2	系统性能评价	15
4.2.1	系统的可靠性	15
4.2.2	系统的可维护性	15
4.2.3	系统的方便性	15
4.2.4	系统中可能存在的隐患	15
5	总 结	16
	参考文献	17

1 引言

1.1 系统开发背景

餐饮业在全国的第三产业中有着举足轻重的作用，一年的零售额达到数千亿元，为中国的税收做出了巨大的贡献。所以餐饮业在中国有着广泛的市场，发展的前景非常好。但是中国的餐饮业也有着自己的发展瓶颈：因为餐饮业的门槛比较低，所以中国的大多数餐饮业老板都是从小店发展起来的，而且家族式的管理比较多。许多还是“人治”，中国的餐饮业没有一套现代企业制度和监督管理体制。更糟的是信息化管理还远远不及外国的先进同行，导致中国的餐饮业在市场竞争中处于劣势。中国的餐饮业要想在国际市场占有一席之地，就必须要提高自己的竞争力，改善自己的经营方式和制度。近年来，随着餐馆规模的逐渐扩大，人工书写的方式已经不能满足如此庞大的数据。为了更好的适应信息时代的高效性，一个利用计算机来实现餐馆管理工作的系统将必然诞生。

1.2 研究意义

现代化的计算机信息管理系统在众多酒店及娱乐服务业已得到广泛应用，满足了广大经营者的迫切需求，它不仅进一步完善了企业管理信息化的进程，更为企业管理走向科学化、规范化、网络化奠定了基础。根据大量用户的实际应用表明，采用计算机管理企业即可节省人力，又起到提高服务速度及准确率作用，同时还可提升企业形象和信誉，提高顾客满意度，大大增强企业的竞争能力和管理水平，综合效益非常明显。计算机辅助点餐管理，可以极大地增强管理者采集、处理信息的能力，从而有利于管理者及时决策；计算机系统能根据管理过程的变化情况，将原始数据、资料等进行加工保存，管理人员可以在解决具体问题需要信息资料时，随时进行检索查询，了解整个点餐管理系统的动态情况，进行动态管理，从而有效的处理点餐的管理工作，实现餐馆管理信息管理的自动化；采用计算机进行管理，实现前、后台服务一次性结账，使餐馆管理条理化、规范化、科学化，极大地提高餐馆管理水平和工作效率。作为日益发展的饮食服务行业，着眼于未来，餐馆采用计算机辅助管理，是与现代化餐馆管理模式接轨的明智选择。

1.3 国内外研究现状

国内已经有不少公司开发出电子点菜系统，例如上海工理电子有限公司的点菜通、餐饮通等，北京辰森计算机系统有限公司的手持点菜宝等。有些大型的餐饮企业已经采用了这两个公司的产品。但是由于成本等原因使用电子点菜系统的餐馆还比较少，市场需求很大。大多数餐饮管理系统可以帮助餐饮企业对企业运行中大量的、动态的、错综复杂的数据和信息进行及时、准确的分析和处理，及时反馈给管理层。采用当今餐饮管

理系统，将使您的企业紧跟科技信息时代的步伐，在行业中的发展中占得先机。

1.4 关键技术分析

本系统数据存储使用本地的 MySQL 数据库并用 Python 程序设计语言来实现。通过使用 SQLite3 相关技术来连接数据库，实现从数据库中查询、插入、更新、删除数据。系统页面采用 Pyqt5、Pyqt5-tools 等 GUI 相关技术，用以提示操作、显示操作的结果等，也可通过数据库可视化工具 Navicat 来查看操作的结果。

1.4.1 SQLite5

SQLite5 是一款轻量级的、基于文件的嵌入式数据库，实现自包容、零配置、支持事务的 SQL 数据库引擎。与其他数据库管理系统不同，SQLite5 的安装和运行非常简单，在大多数情况下，只要确保 SQLite5 的二进制文件存在即可开始创建、连接和使用数据库。第一个 Alpha 版本诞生于 2000 年 5 月，直到今天已经成为最流行的嵌入式数据库，包括 Google 在内的许多公司在其桌面软件中亦使用 SQLite5 存储用户数据，由此看来，其稳定性毋庸置疑。

1.4.2 PyQt5、PyQt5-tools

PyQt5 是一套 Python 绑定 Digia QT5 应用的框架 Q。它可用于 Python。Qt 库是最强大的 GUI 库之一。PyQt5 的官方网站 Riverbank Computing News。

PyQt5 作为 Python 的一个模块，它有 620 多个类和 6000 个函数和方法。这是一个跨平台的工具包，它可以运行在所有主要的操作系统，包括 UNIX, Windows, Mac OS。PyQt5 是双重许可。开发者可以在 GPL 和商业许可之间进行选择。其中 PyQt5-tools 为 Qt Designer 拖拽式的界面设计工具，为开发人员省去大量设计页面时间。在 PyQt 中编写 U 界面可以直接通过代码来实现，也可以通过 Qt Designer 来完成。Qt Designer 的设计符合 MVC 的架构，其实现了视图和逻辑的分离，从而实现了开发的便捷。Qt Designer 中的操作方式十分灵活，其通过拖拽的方式放置控件可以随时查看控件效果。QtDesigner 生成的.ui 文件(实质上是 XML 格式的文件)也可以通过 pyuic5 工具转换成.py 文件。

Qt Designer 随 PyQt5-tools 包一起安装，若要启动 Qt Designer 可以直接到上述目录下，双击 designer.exe 打开 Qt Designer;或将上述路径加入环境变量，在命令行输入 designer 打开;或在 PyCharm 中将其配置为外部工具打开。

因为可用的类有很多，他们被分成几个模块。QtCore 模块包含核心的非 GUI 功能。该模块用于时间、文件和目录、各种数据类型、流、网址、MIME 类型、线程或进程。QtGui 模块包含图形组件和相关的类，例如按钮、窗体、状态栏、工具栏、滚动条、位图、颜色、字体等。QtNetwork 模块包含了网络编程的类，这些类允许编写 TCP/IP 和 UDP 的客户端和服务端，他们使网络编程更简单，更轻便。QtXml 包含使用 XML 文件的类，这个模块提供了 SAX 和 DOM API 的实现。QtSvg 模块提供显示的 SVG 文件的类。可缩放矢量

图形（SVG）是一种用于描述二维图形和图形应用程序的 XML 语言。QtOpenGL 模块使用 OpenGL 库渲染 3D 和 2D 图形，该模块能够无缝集成 Qt 的 GUI 库和 OpenGL 库。QtSql 模块提供用于数据库的类。本次课程设计主要使用了 PyQt5 中的 QtGui 类。

1.5 系统的特色和创新之处

目前，现阶段市场上所呈现出的餐馆收银系统，绝大部分是为大型的餐馆企业进行设计的，优点是其管理功能普遍较为全面，但是其存在的不足也是较为明显的：许多功能并不适用于中小型的餐馆企业。实际上，在当今国内市场中占据份额比例更大一块的却是中小型餐馆企业，与之对应的却是适用于中小型企业的餐馆收银系统数量非常少。因此，该系统主要是针对中小型企业的餐馆而开发的管理系统。

1.6 本章小结

本章主要介绍课题的开发背景和研究意义，探讨该课题的国内外研究现状，分析本系统所使用到的相关技术，以及本系统的特色创新之处。

2 需求分析

2.1 可行性分析

系统的可行性分析主要从系统实施的影响、技术可行性、经济可行性、操作可行性以及其他方面因素进行阐述和分析。

2.1.1 系统实施的影响分析

对于中小餐馆企业来说，餐馆收银系统首当其冲的就是提高了服务的质量。其次，中小型餐馆使用餐馆收银软件可以加强餐馆管理和提高工作效率，进而提高餐馆的经济效益。

2.1.2 技术可行性

开发一个中小型的餐馆收银系统，所涉及到的技术问题不会太多，主要用到的工具就是数据库管理系统和一门可视化开发的编程语言。MySQL 体积小、速度快、总体拥有成本低、开源；支持多种操作系统；是开源数据库，提供的接口支持多种语言连接操作。而本系统采用的开发工具集成开发环境中的 Python 语言自身所提供的各种控件、函数及方法完全可以帮助我们完成系统的开发。在实际的开发过程和应用中，数据库和应用程序均在同一计算机上运行。数据库中储存了所需要被处理的数据，并由 MySQL 数据库管理系统对其进行集中管理，应用程序由 PyQt5-tools 集成开发环境中的 GUI 控件进行可视化开发，这为系统的调试工作带来了便利。

2.1.3 经济可行性

该餐馆收银系统从开发、使用到维护所需要的费用都十分的低廉，且在软硬件方面的要求都不是很高，而收益却非常的巨大，对于中小型的餐馆企业来说，不但不会增加营业成本，反而能够提高自身的管理效率，创造更多的经济利益。

2.1.4 操作可行性

系统设计采用模块化结构，整体系统是由多个子系统构成，各个系统子模块相对独立，系统操作采用会话式处理，操作员可根据提示信息操作，易于掌握，不易出错，有初中文化程度经过短期训练的操作员即可使用。中小型餐馆一个人可以完成总服务台的几乎全部工作，大大节约了人力成本。

2.1.5 其他因素分析

1) 社会

该系统如果中小型餐馆使用得当，可以很好的占领客户市场，从而占取更多的市场份额，经济得到良好的发展，因此本系统对于社会主要就是经济发展方面的影响，不存在其他的重大影响。

2) 健康

餐馆收银系统可以大大提高管理的效率，缩短工作时间，不再像以前手工录入时较劳累，但借助该系统进行管理需要通过使用电脑或其他电子产品来进行一系列的操作，而电子产品一般辐射都比较大，对身体有一定伤害。

3) 法律

首先，本系统所管理的内容完全是符合社会主义要求的，不存在反国反党的言论或者信息；其次，本系统内的信息是十分健康的，不存在辱骂或者脏话的信息；最后，本系统不存在侵权问题。

4) 文化

本系统对企业管理文化有推动的作用，会逐渐覆盖传统的餐馆收银模式，推进了餐馆管理文化向着信息化发展。

5) 环境

本系统完全在电脑端操作，不会产生任何对环境有害的污染。

2.2 总体架构

本系统的用户是餐馆的管理员，根据生成的菜单操作，有利于工作的区域承担。主要的流程就是进入系统后，对数据库的数据进行操作，修改数据库中的数据，然后页面的数据会根据数据库的操作改变。本系统的总体架构图如表 2.1 所示。

表 2.1 系统的总体架构表

用户	餐馆管理员
表现层	GUI
应用层	餐馆收银系统
数据层	MySQL 数据库

2.3 功能需求分析

2.3.1 需求描述

分析功能需求需要从系统的使用用户角度考虑到各种方面的问题，要求整个系统满足所有的需求，具有实际操作的意义，足够人性化，操作简单。可以确定的是本系统的用户是餐馆的管理员，从这类用户的角度出发，考虑本系统的各方面需求。

1. 会员管理：包括注册会员和换绑手机。

(1) 注册会员。管理员通过输入会员编号，姓名，手机号实现注册。

(2) 换绑手机。管理员通过输入旧手机号换绑新手机号。

2. 菜单管理：包括添加菜品，删除菜品和修改菜价。

(1) 添加菜品。管理员通过输入菜品编号，菜品名称，菜品价格添加菜品。

- (2) 删除菜品。管理员通过输入菜品名称删除菜品。
- (3) 修改菜价。管理员通过输入菜品名称和新的菜价来修改菜价。
- 3. 账单结算：管理员通过输入会员手机号，菜品名称和消费时间来实现加菜账单记录。
- 4. 打印小票：管理员通过输入会员手机号和消费时间来打印小票内容。

2.3.2 数据流图

顶层数据流图如图 2.2 所示。

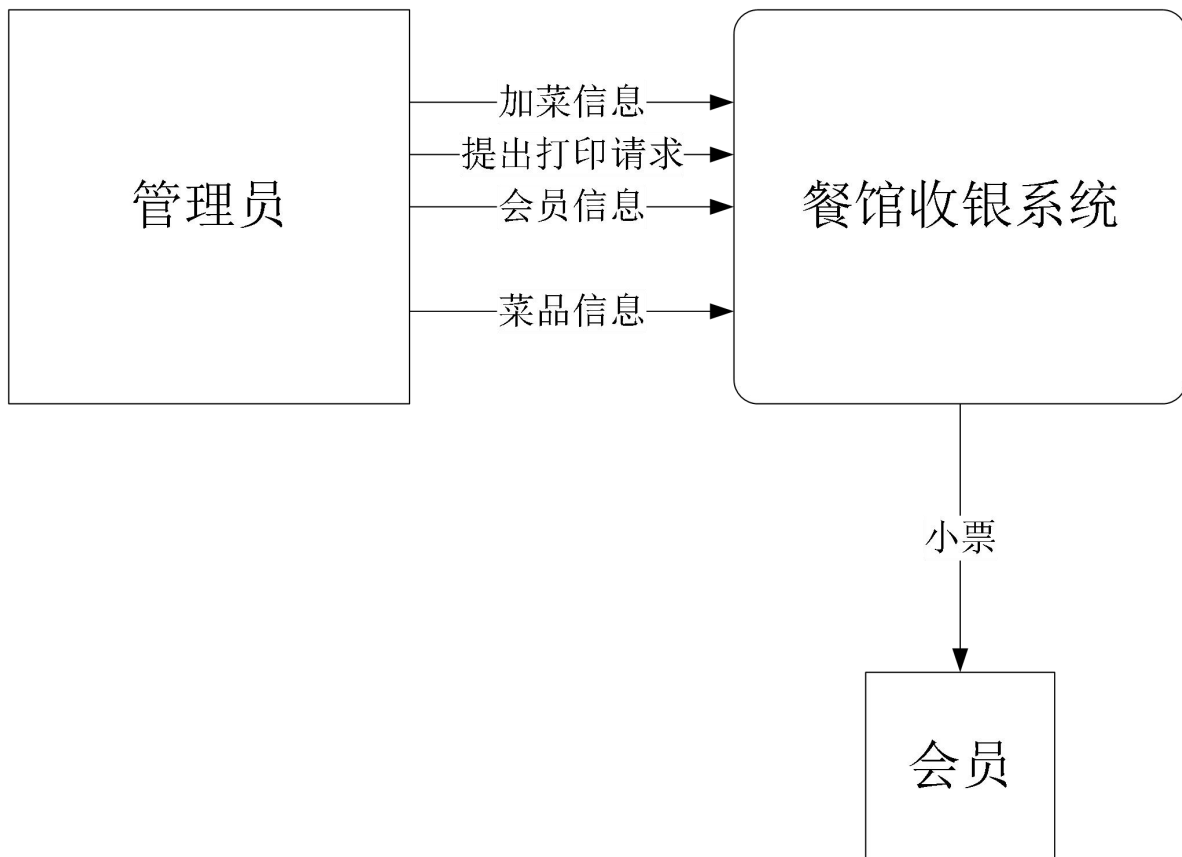


图 2.2 顶层数据流图

零层数据流图如图 2.3 所示。

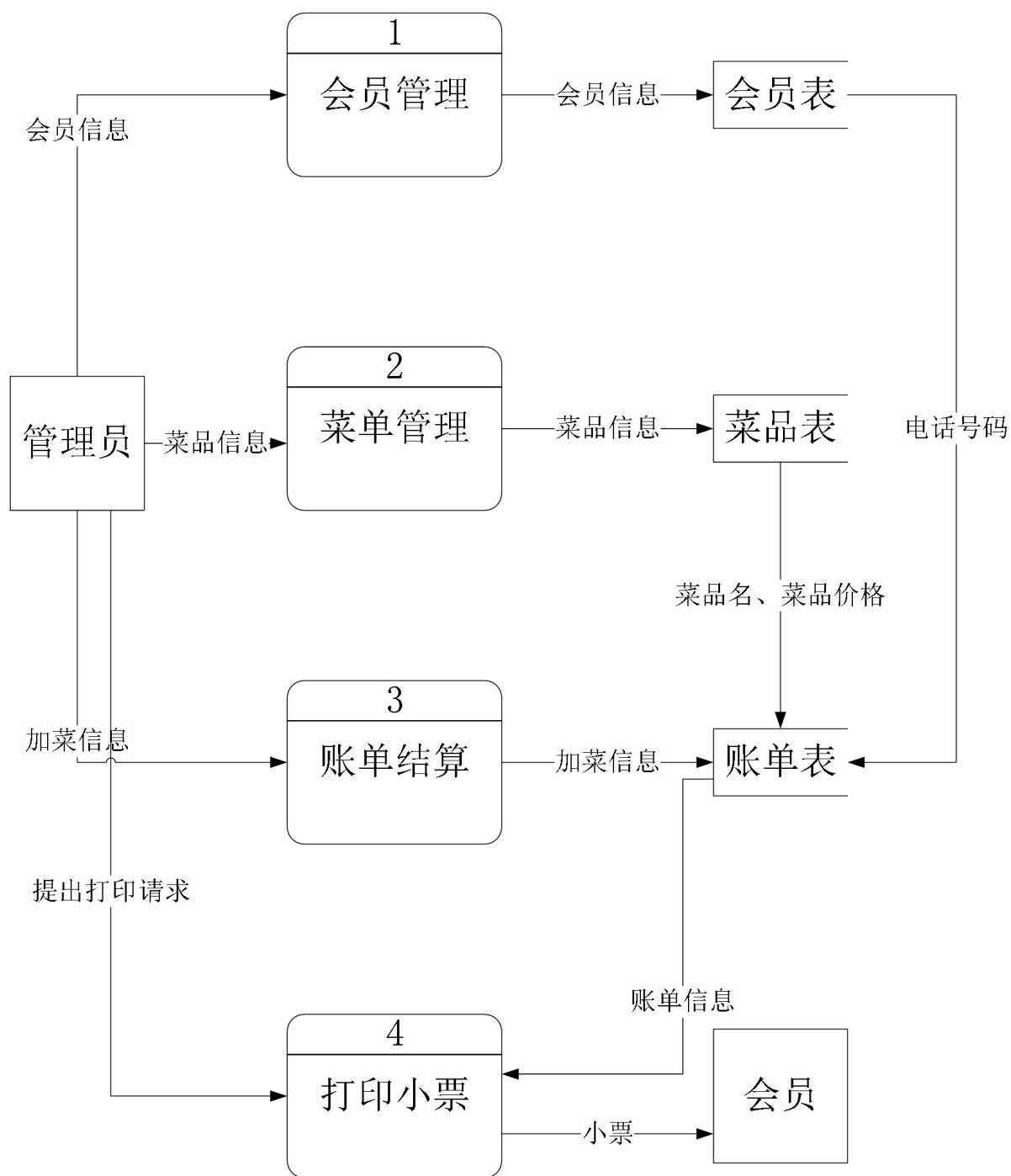


图 2.3 零层数据流图

2.3.3 数据字典

会员信息 = 会员编号 + 会员名 + 电话号码

菜品信息 = 菜品编号 + 菜品名 + 菜品价格

加菜信息 = 菜品名 + 电话号码 + 点餐时间

提出打印申请 = 电话号码 + 点餐时间

账单信息 = 菜品名 + 菜品价格

小票 = 菜品名 + 菜品价格 + 总价

数据项列表如表 2.4 所示。

表 2.4 数据项列表

数据项名	数据项含义	数据类型	长度
会员编号	唯一标识会员	文本	255
会员名	会员的名字	文本	255
电话号码	会员的电话号码	文本	255
菜品编号	唯一标识菜品	文本	255
菜品名称	菜品的名字	文本	255
菜品价格	菜品的价格	小数	10
账单菜品	账单的菜品名称	文本	255
账单价格	账单的菜品价格	小数	10
账单电话	账单的电话号	文本	255
账单日期	账单发生的日期	文本	255

2.4 系统的非功能性需求

分析系统的非功能性需求进，主要从系统性能容量、系统维护、安全等方面考虑。

2.4.1 性能、容量

本系统开发使用的软件以及技术，都是以简便、快速为目的，所以性能方面是不错的，反应速度也较快。系统使用的 MySQL 数据库，容量足以承受中小型企业的信息量需求。

2.4.2 可靠性

系统开发使用的技术和软件都是被普遍使用的，不会轻易崩溃导致系统无法运行。同时，对数据的管理是严格的。

2.4.3 可维护性

本系统一般时不需要进行维护的，只需要根据今后需要的功能进行增改即可。

2.4.4 安全

本系统为简单的点菜记账、打印小票、记录账本等功能，较追求实时性，即数据即出即用，安全方面不需较大关心。

2.5 本章小结

本章主要是对系统的功能性以及非功能性需求的各方面分析。

3 系统设计

3.1 系统设计

3.1.1 功能模块图

功能模块图如图 3.1 所示。

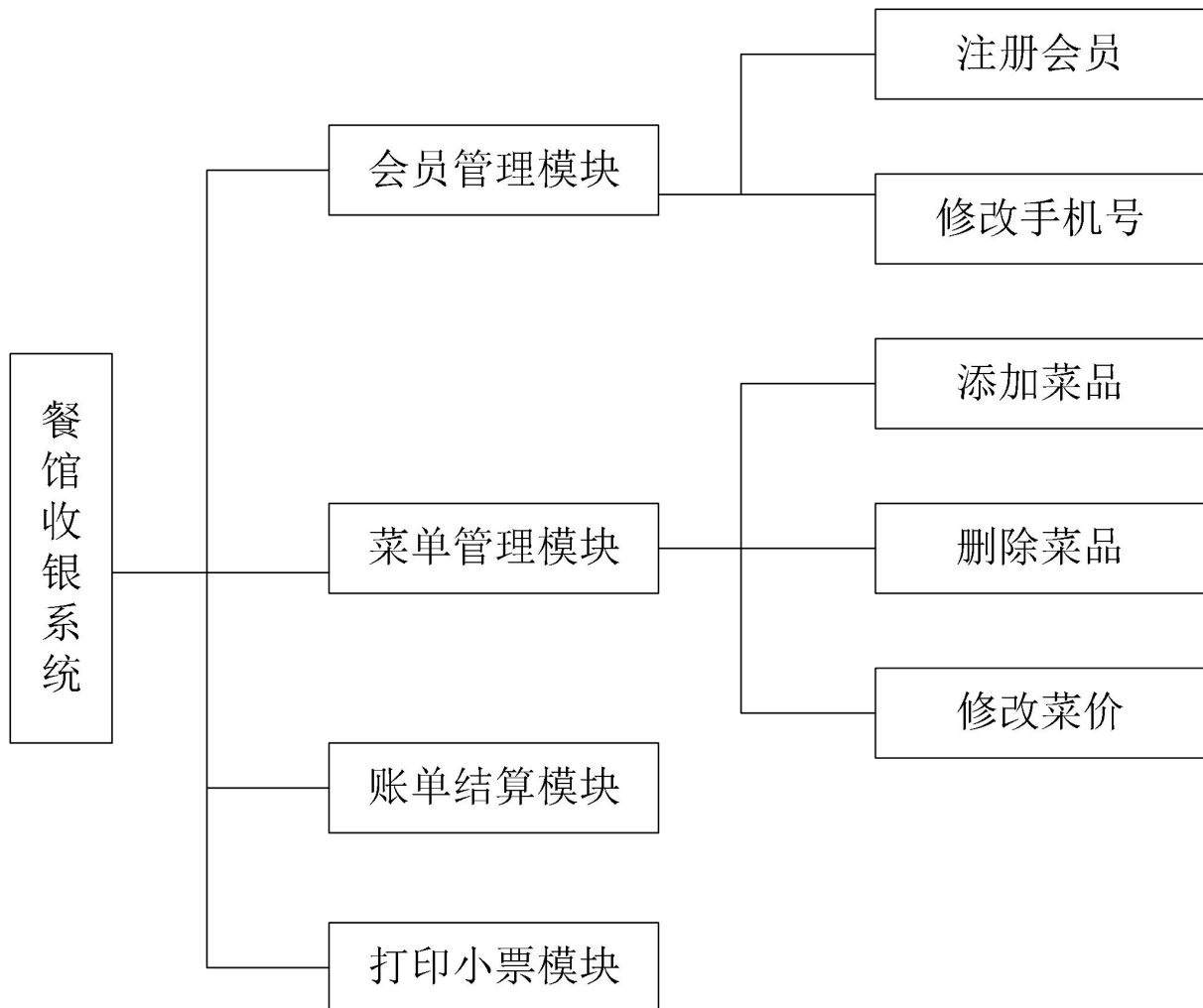


图 3.1 功能模块图

3.1.2 系统功能分析

注册会员时，管理员通过输入会员编号，姓名，手机号实现注册。

换绑手机时，管理员通过输入旧手机号换绑新手机号。

添加菜品时，管理员通过输入菜品编号，菜品名称，菜品价格添加菜品。

删除菜品时，管理员通过输入菜品名称删除菜品。

修改菜价时，管理员通过输入菜品名称和新的菜价来修改菜价。

账单结算时，管理员通过输入会员手机号，菜品名称和消费时间来实现加菜账单记录。

打印小票时，管理员通过输入会员手机号和消费时间来打印小票内容。

3.1.3 系统用例图等

系统用例图如图 3.2 所示。

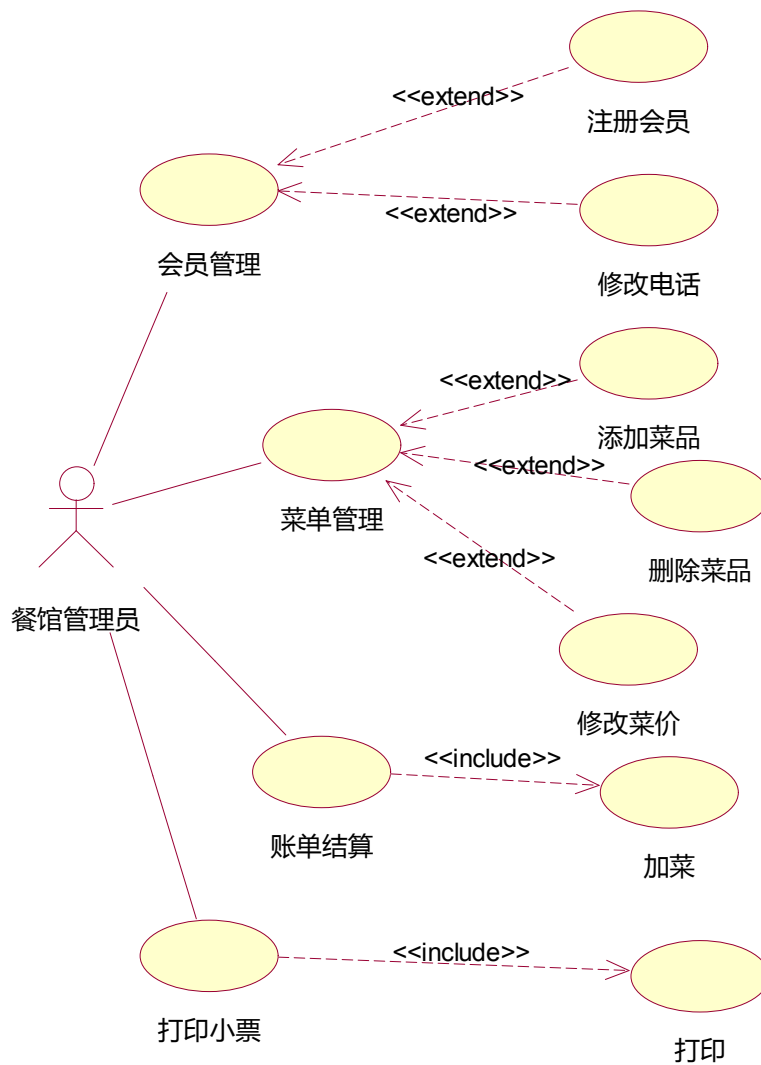


图 3.2 系统用例图

3.2 数据库设计

3.2.1 局部概念模式设计

会员实体的属性图如图 3.3 所示。

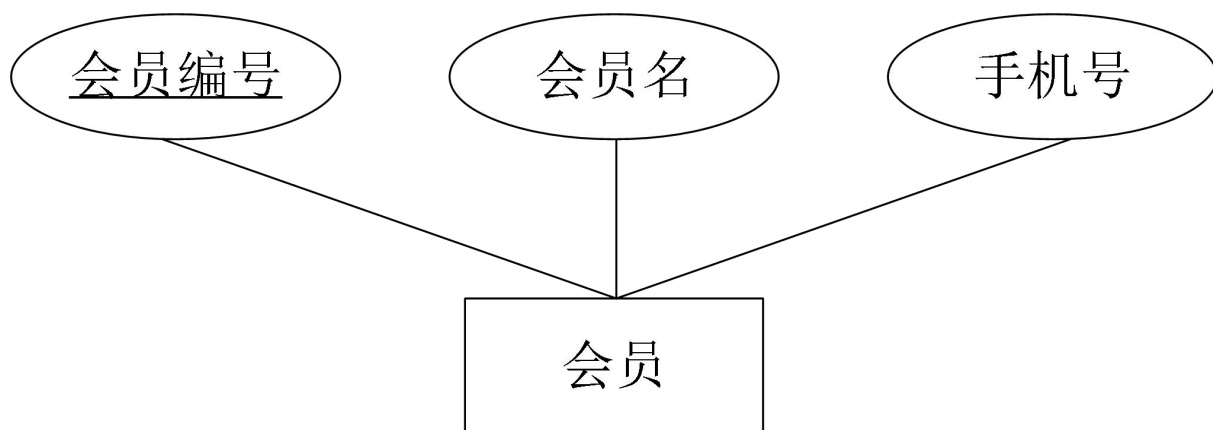


图 3.3 会员属性图

菜单实体的属性图如图 3.4 所示。

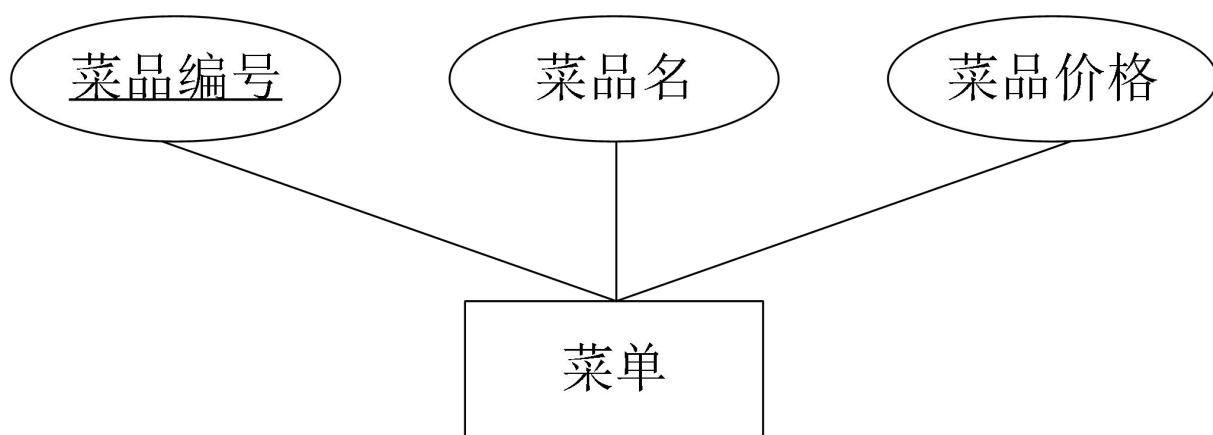


图 3.4 菜单属性图

账单实体的属性图如图 3.5 所示。

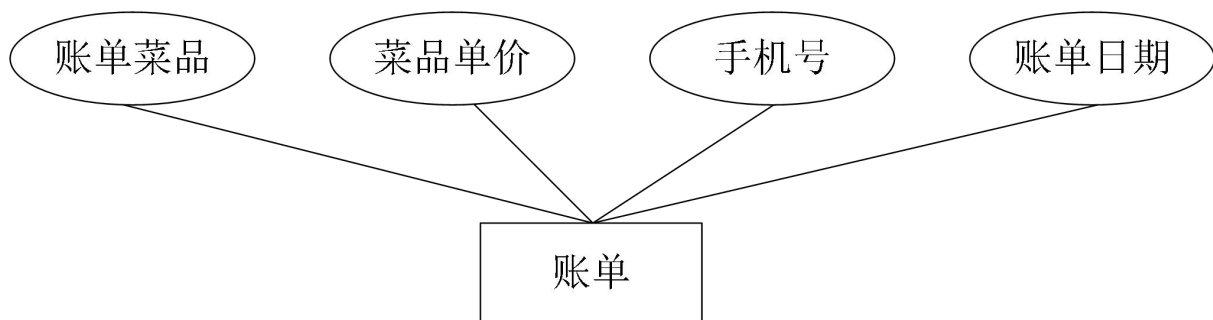


图 3.5 账单属性图

3.2.2 全局概念模式设计

全局概念模式设计图如图 3.6 所示。

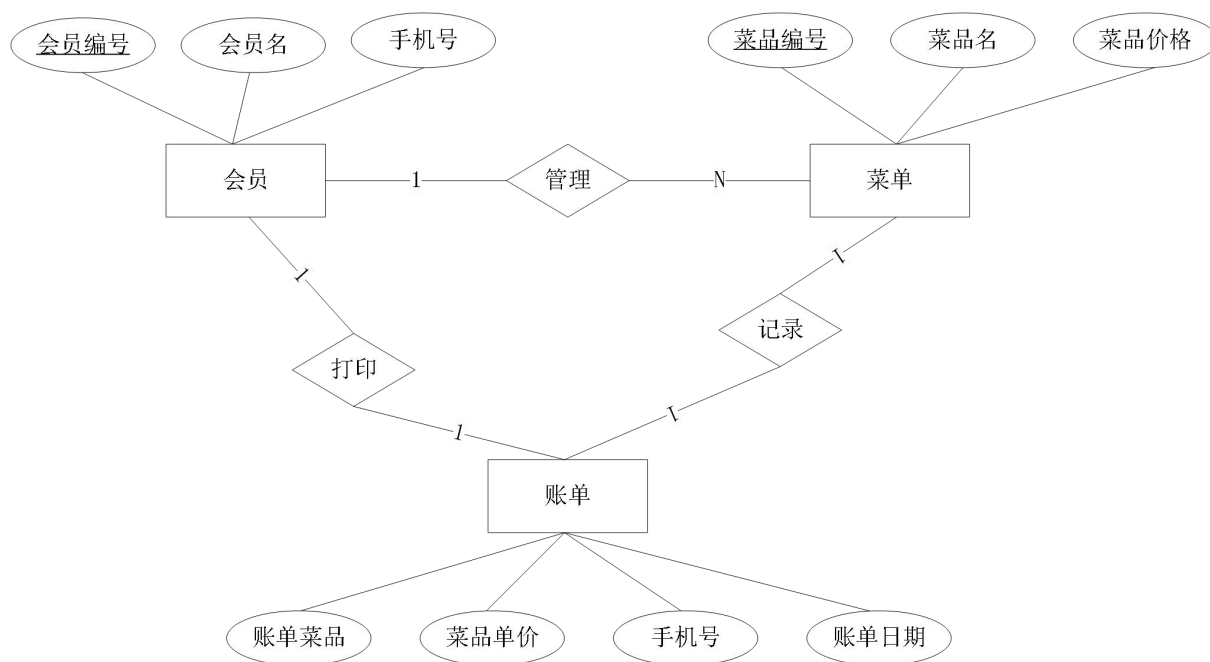


图 3.6 全局概念模式设计图

3.2.3 数据库设计

会员信息表（会员编号，会员名，手机号）

菜单信息表（菜品编号，菜品名，菜品价格）

账单信息表（账单菜品，菜品单价，手机号，账单日期）

账单信息表没有设计主键，其原因为中小型餐馆账单流水小，不必特意设置账单编号，只需通过当日账单的日期和对应的会员手机号获取该会员本次消费即可打印小票，虽然其记录信息都有冗余。

属性列表如表 3.7 所示。

表 3.7 属性列表

属性名	属性含义	数据类型	长度
会员编号	唯一标识会员	文本	255
会员名	会员的名字	文本	255
电话号码	会员的电话号码	文本	255
菜品编号	唯一标识菜品	文本	255
菜品名称	菜品的名字	文本	255
菜品价格	菜品的价格	小数	10
账单菜品	账单的菜品名称	文本	255
账单价格	账单的菜品价格	小数	10
账单电话	账单的电话号	文本	255
账单日期	账单发生的日期	文本	255

3.3 本章小结

本章主要分析系统的功能模块，实体之间的关系以及实体的属性，设计 E-R 图，然后根据 E-R 图，设计数据库，并做出数据库表。

4 系统设计与实现

4.1 系统相关功能的实现

4.1.1 注册会员

系统提示管理员需要输入会员的对应信息，再将输入的会员信息写入数据库中。具体实现流程图如图 4.1 所示。

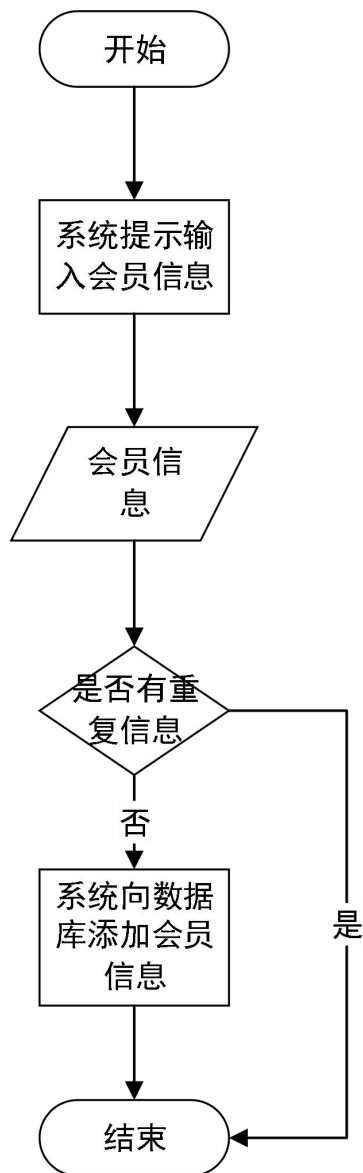


图 4.1 注册会员流程图

在菜单界面输入注册会员的信息后，点击注册按钮，若数据库中没有重复信息系统给予操作成功提示框，若存在重复信息，则数据库中没有该信息。注册会员的页面效果如图 4.2 所示。



图 4.2 注册会员效果图

注册会员的后端相关代码：

#会员注册功能

```
def memberRegister(self):
    conn = get_conn()
    cur = conn.cursor()
    sql = "insert into member(memberId, memberName, memberPhone) values(%s,%s,%s)"
    memberId = self.memberId.text()
    memberName = self.memberName.text()
    memberPhone = self.memberPhone.text()
    cur.execute(sql, (memberId, memberName, memberPhone))
    conn.commit()
    close_conn(conn, cur)
    self.success_app = SuccessApp()
    self.success_app.show()}
```

4.1.2 换绑手机

系统提示管理员需要输入会员的旧手机号，再将输入的新手机号。具体实现流程图如图 4.3 所示。

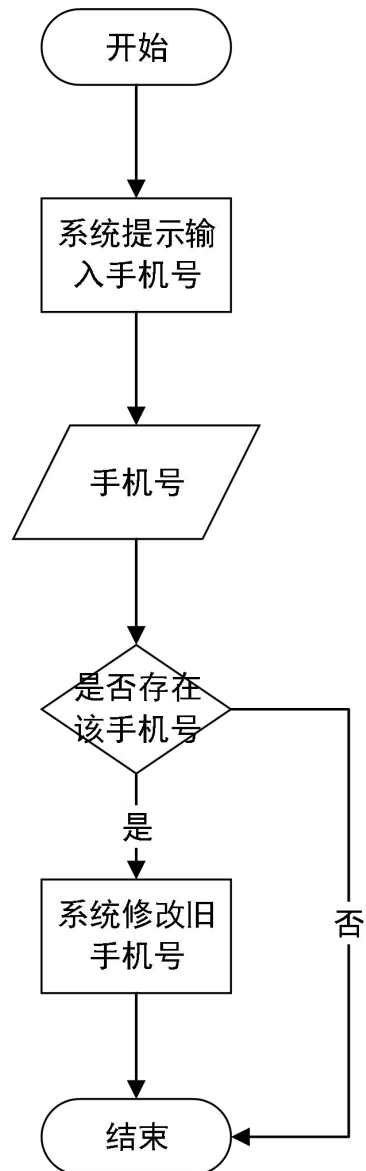


图 4.3 换绑手机流程图

在菜单界面输入会员的旧手机号后，点击修改按钮，若数据库中有该信息，系统给予操作成功提示框，若不存在该信息，则数据库不受影响。注册会员的页面效果如图 4.4 所示。



图 4.4 换绑手机效果图

换绑手机的后端相关代码：

#换绑手机功能

```
def phoneUpdate(self):  
    conn = get_conn()  
    cur = conn.cursor()  
    sql = "update member set memberPhone=%s where memberPhone=%s"  
    oldPhone = self.oldPhone.text()  
    newPhone = self.newPhone.text()  
    cur.execute(sql, (newPhone, oldPhone))  
    conn.commit()  
    close_conn(conn, cur)  
    self.success_app = SuccessApp()  
    self.success_app.show()
```

4.1.3 添加菜品

系统提示管理员需要输入菜品编号、菜品名、菜品价格。具体实现流程图如图 4.5 所示。

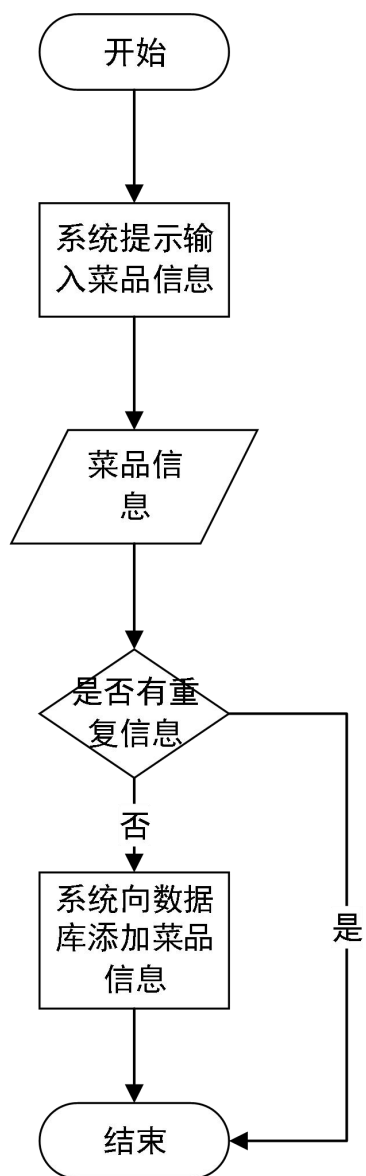


图 4.5 添加菜品流程图

在菜单界面输入菜品编号、菜品名、菜品价格后，点击添加按钮，若数据库中有该信息，系统给予操作成功提示框，若不存在该信息，则数据库不受影响。添加菜品的页面效果如图 4.6 所示。

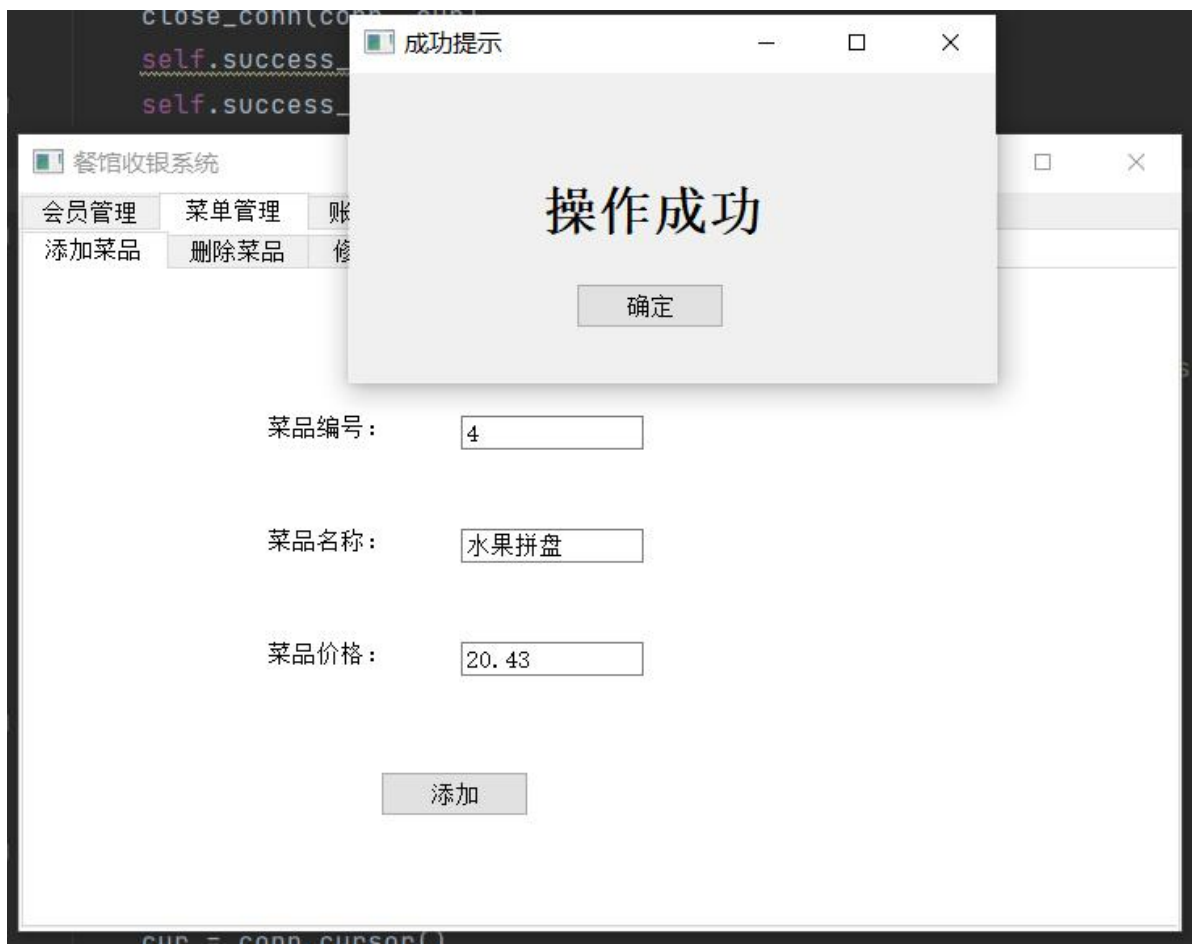


图 4.6 添加菜品效果图

添加菜品的后端相关代码：

#添加菜品功能

```
def menuAdd(self):
    conn = get_conn()
    cur = conn.cursor()
    sql = "insert into menu(menuId, menuDish, menuPrice) values(%s,%s,%s)"
    menuId = self.menuId.text()
    menuDish = self.menuDish.text()
    menuPrice = self.menuPrice.text()
    cur.execute(sql, (menuId, menuDish, menuPrice))
    conn.commit()
    close_conn(conn, cur)
    self.success_app = SuccessApp()
    self.success_app.show()
```

4.1.4 删除菜品

系统提示管理员需要输入菜品名。具体实现流程图如图 4.7 所示。

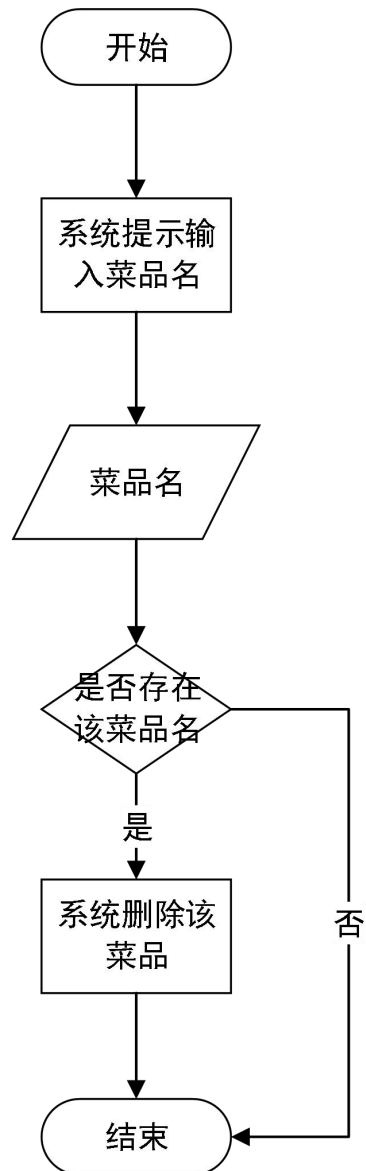


图 4.7 删除菜品流程图

在菜单界面输入菜品名后，点击删除按钮，若数据库中有该信息，系统给予操作成功提示框，若不存在该信息，则数据库不受影响。删除菜品的页面效果如图 4.8 所示。

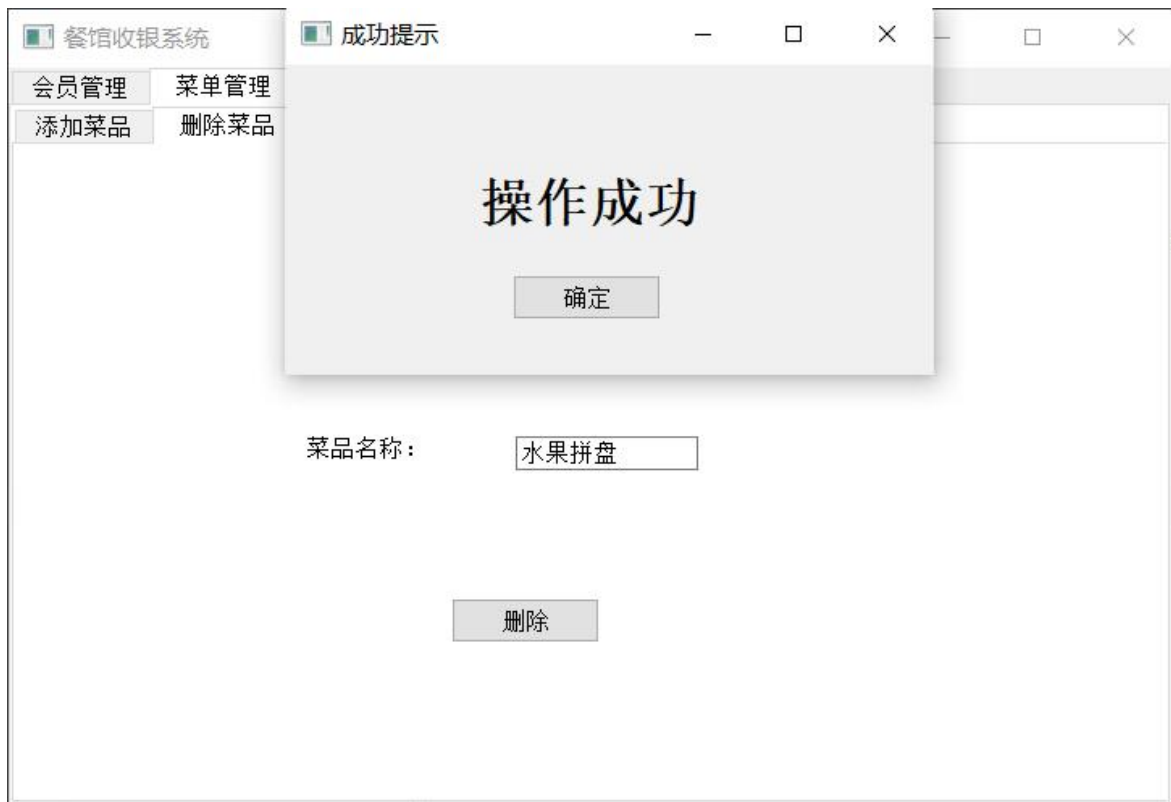


图 4.8 删除菜品效果图

删除菜品的后端相关代码：

#删除菜品功能

```
def menuDelete(self):  
    conn = get_conn()  
    cur = conn.cursor()  
    sql = "delete from menu where menuDish=%s"  
    menuDish = self.menuDish_2.text()  
    cur.execute(sql, (menuDish,))  
    conn.commit()  
    close_conn(conn, cur)  
    self.success_app = SuccessApp()  
    self.success_app.show()
```

4.1.5 修改菜价

系统提示管理员需要输入菜品名、新的菜品价格。具体实现流程图如图 4.9 所示。

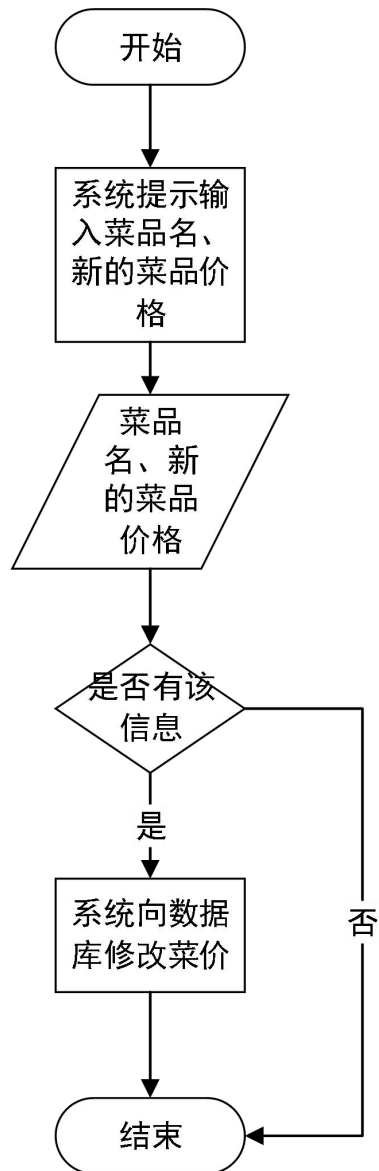


图 4.9 修改菜价流程图

在菜单界面输入菜品名、新的菜品价格后，点击修改按钮，若数据库中有该信息系统给予操作成功提示框，若不存在该信息，则数据库不受影响。添加菜品的页面效果如图 4.10 所示。

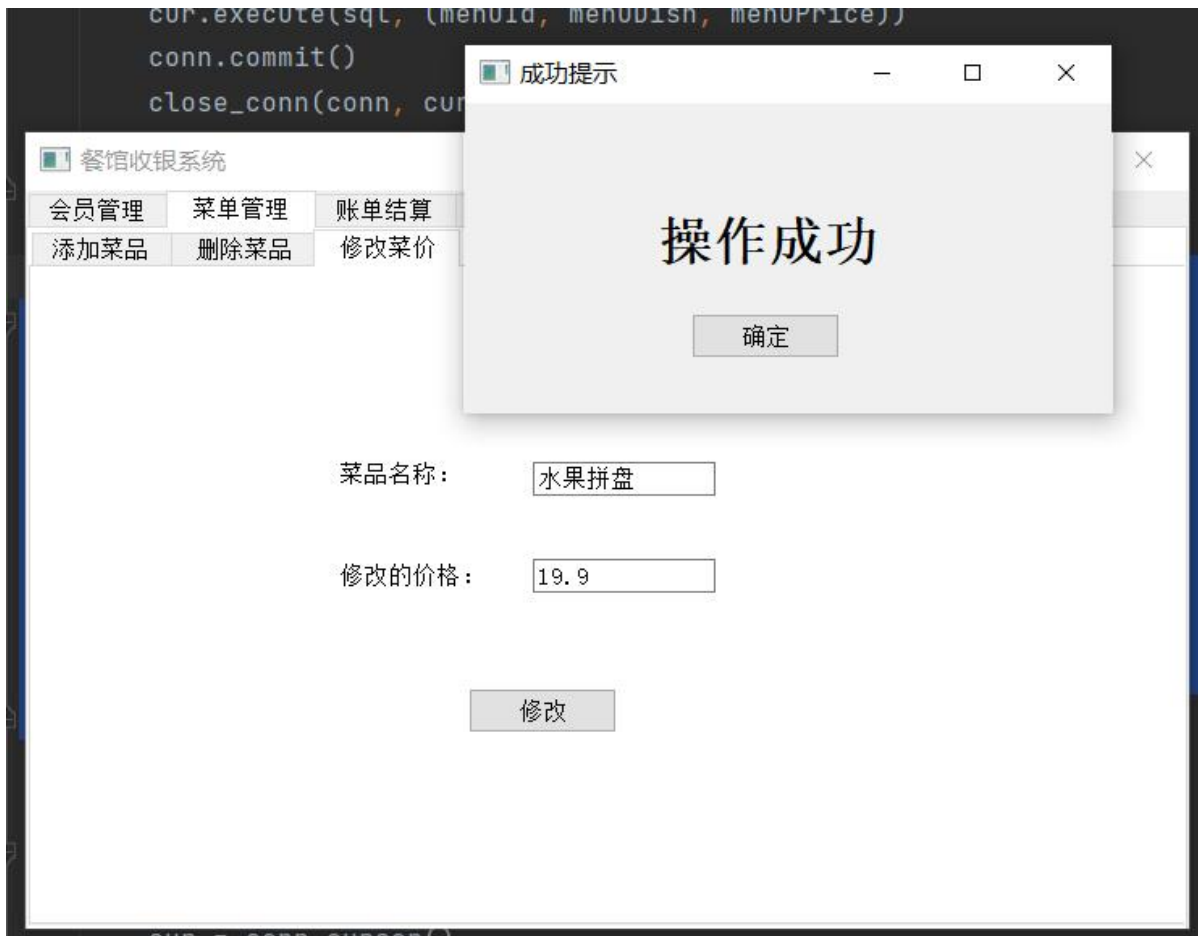


图 4.10 修改菜价效果图

修改菜价的后端相关代码:

#修改菜价功能

```
def menuUpdate(self):  
    conn = get_conn()  
    cur = conn.cursor()  
    sql = "update menu set menuPrice=%s where menuDish=%s"  
    menuPrice = self.menuPrice_2.text()  
    menuDish = self.menuDish_3.text()  
    cur.execute(sql, (menuPrice, menuDish))  
    conn.commit()  
    close_conn(conn, cur)  
    self.success_app = SuccessApp()  
    self.success_app.show()
```

4.1.6 账单结算-加菜

系统提示管理员需要输入菜品编号、菜品名、菜品价格。具体实现流程图如图 4.11 所示。

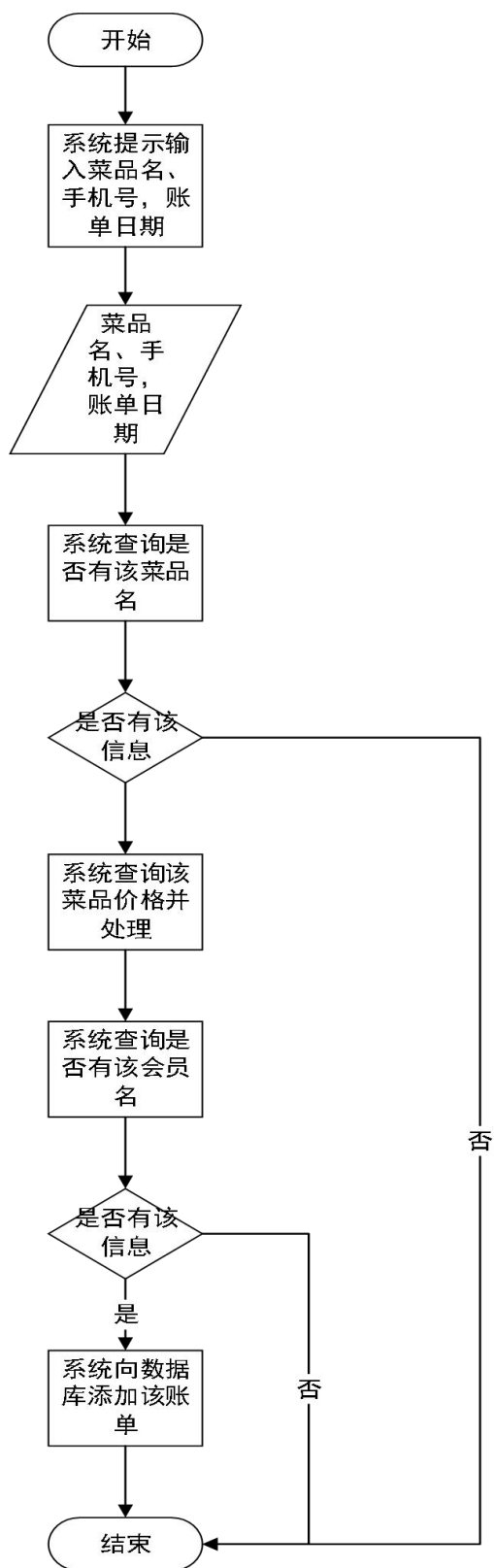


图 4.11 账单结算-加菜流程图

在菜单界面输入菜品编号、菜品名、菜品价格后，点击添加按钮，若数据库中有该信息，则数据库不受影响。添加菜品的页面效果如图 4.12 所示。

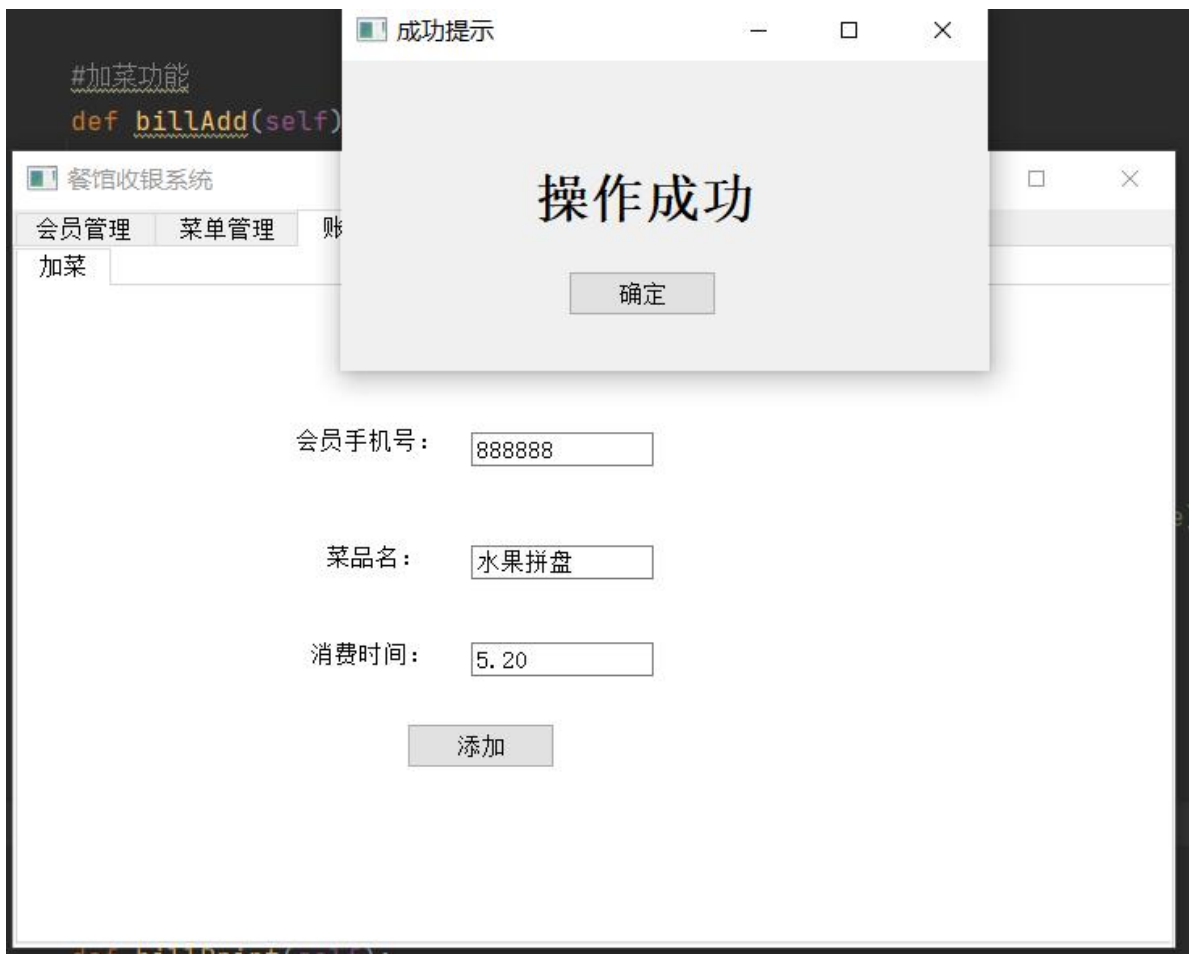


图 4.12 账单结算-加菜效果图

添加菜品的后端相关代码：

#加菜功能

```
def billAdd(self):
    conn = get_conn()
    cur = conn.cursor()
    sql1 = "select menuPrice from menu where menuDish=%s"
    billDish = self.billDish.text()
    cur.execute(sql1, (billDish,))
    menuPrice = cur.fetchall()
    for row in menuPrice:
        billPrice = row[0]
    sql2 = "insert into bill(billDish, billPrice, billPhone, billDate) values(%s,%s,%s,%s)"
    billPhone = self.billPhone.text()
    billDate = self.addDate.text()
    cur.execute(sql2, (billDish, billPrice, billPhone, billDate))
    conn.commit()
```

```
close_conn(conn, cur)
self.success_app = SuccessApp()
self.success_app.show()
```

4.1.7 打印小票

系统提示管理员需要输入会员的手机号和消费时间，通过查询数据库来显示到界面上。具体实现流程图如图 4.13 所示。

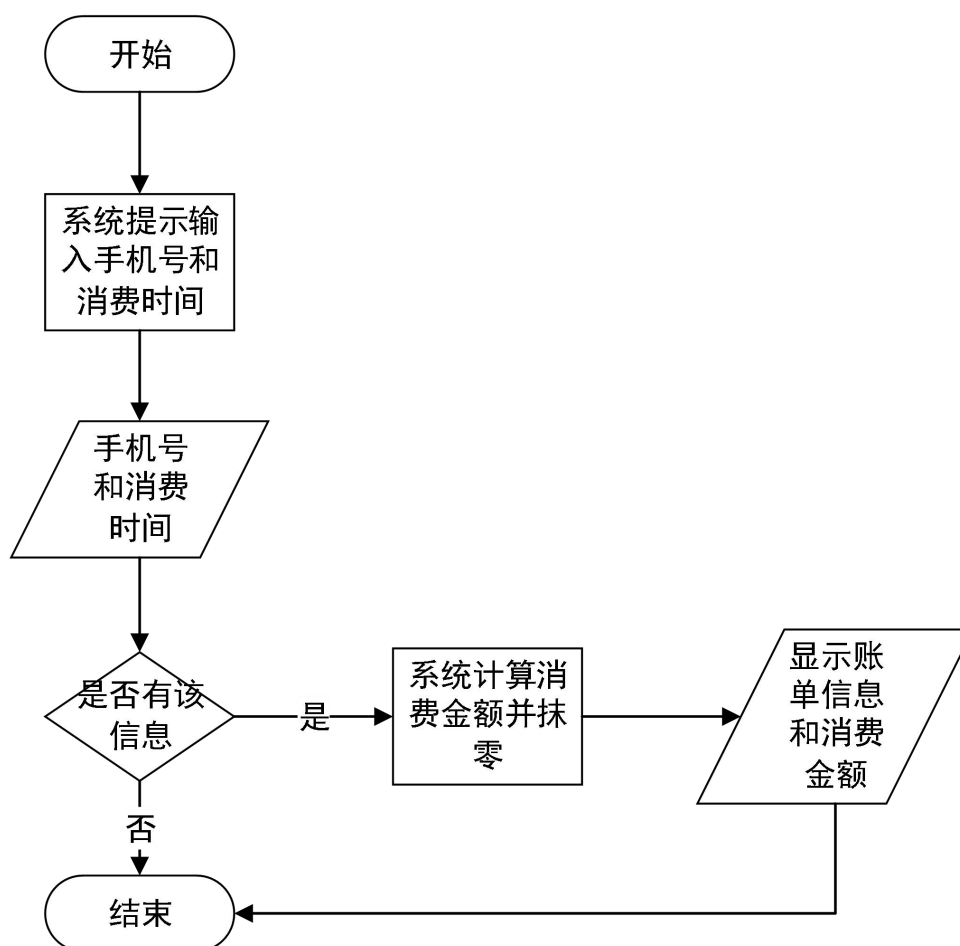


图 4.13 打印小票流程图

在菜单界面输入会员的手机号和消费时间后，点击打印按钮，若数据库中有该账单信息，系统将查询内容显示到界面上，同时显示计算的总价，并给予操作成功提示框，若不存在该信息，则不显示任何信息，金额为 0。注册会员的页面效果如图 4.14 所示。

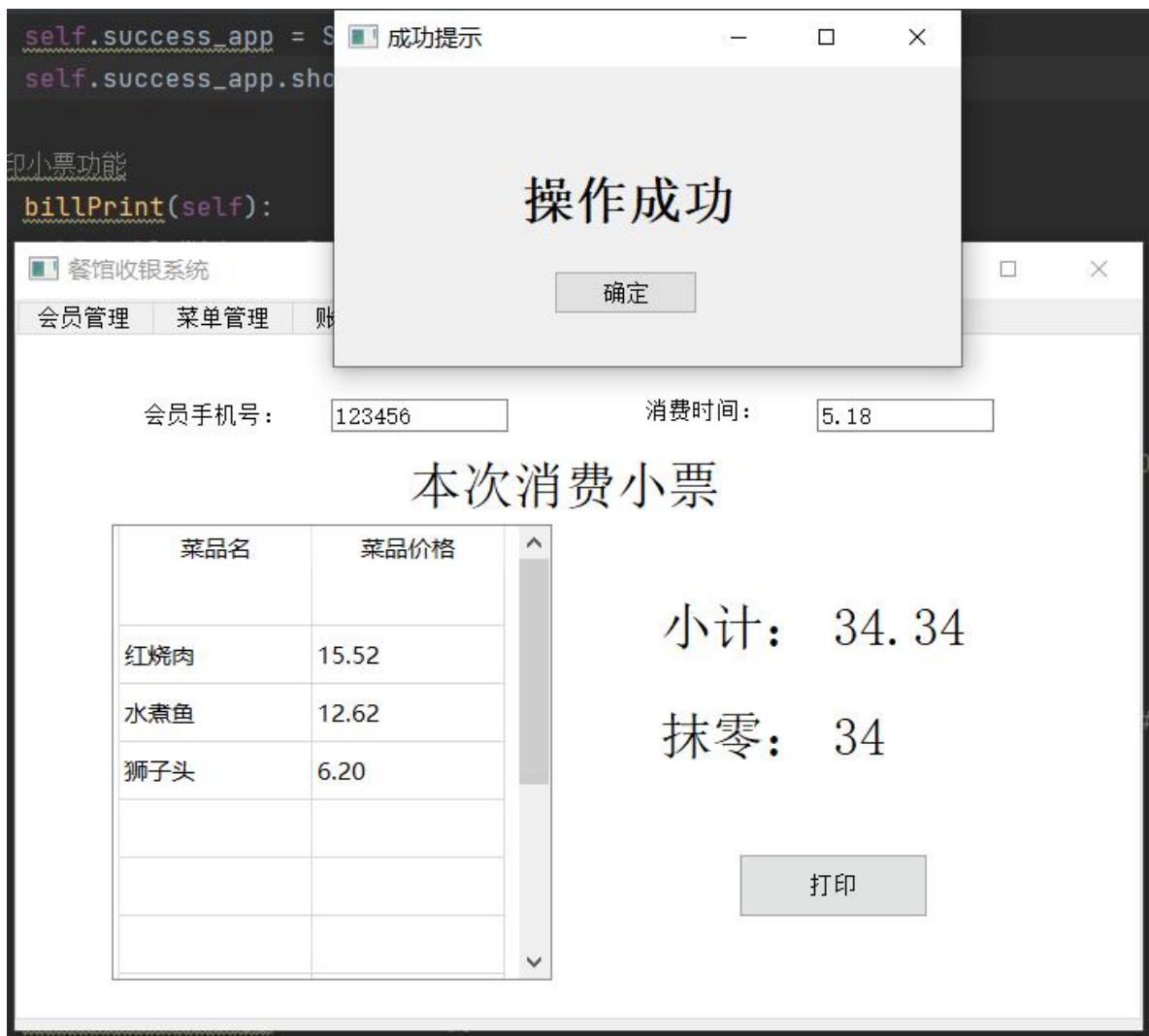


图 4.14 打印小票效果图

打印小票的后端相关代码:

#打印小票功能

```
def billPrint(self):
```

```
    self.tableWidget.clearContents(); #清空表格内容
```

```
    s = 0
```

```
    count = 1
```

```
    conn = get_conn()
```

```
    cur = conn.cursor()
```

```
    sql1 = "select billDish,billPrice from bill where billPhone=%s and billDate=%s"
```

```
    billPhone = self.printPhone.text()
```

```
    billDate = self.billDate.text()
```

```
    cur.execute(sql1, (billPhone,billDate))
```

```
    bill = cur.fetchall()
```

```
    for row in bill:
```

```
        self.tableWidget.setItem(count, 0, QTableWidgetItem(str(row[0]))) # 列表填写内
```

容, 参数含义 (行数, 列数, 内容)

```
        self.tableWidget.setItem(count, 1, QTableWidgetItem(str(row[1])))
```

```
        s += row[1]
```

```
count += 1
self.sum.setText(str(s))
self.result.setText(str(int(s)))
close_conn(conn, cur)
self.success_app = SuccessApp()
self.success_app.show()
```

4.2 系统性能评价

4.2.1 系统的可靠性

此系统具有增加、删除、修改、查找、打印小票等操作，可以对餐馆的信息无差错的记录，能优化餐馆管理的流程，节省了大量的人力，而且便于集中管理，同时也大大的提高餐馆管理的速率。并且系统的操作都是鼠标、可视化界面操作，操作比较简单，对电脑熟练掌握以及有一些使用管理软件的经验的人员都能轻易使用。

4.2.2 系统的可维护性

该系统依赖 Mysql、PyQt5 第三方库、PyQt5-tools 实现，这三种开发工具皆长期稳定存在，不会出现大幅度的更新和改动，有利于后期系统的更新和维护，同时开发组员了解各个功能的实现并在代码中进行了详细的标注，提高了代码的可阅读性，对后续人员的使用和维护提供了方便。

4.2.3 系统的方便性

该系统开发贴合了人机交互的基本理念，让使用者使用时便于操作，使系统最大程度上为人们完成信息管理、服务和处理等功能，系统开发过程中，代码编写人员对程序做到了尽可能的提示，方便用户经行操作。

4.2.4 系统中可能存在的隐患

由于时间有限，本系统没有进行充分的测试，能满足绝大多数的操作需要，但可能会出现意外情况，还需要维护。例如在输入重复的信息时，给予的是操作成功提示，但数据库并不会记录该重复信息，还需要进一步改进。

5 总 结

在我看来,数据库课程设计主要目标是利用课程中学到数据库知识和技术较好开发出数据库应用系统,去解决各行各业信息化处理要求。通过这次课程设计,可以巩固我们对数据库基本原理和基础理论理解,掌握数据库应用系统设计开发基本方法,进一步提高我们综合运用所学知识能力。在这次数据库设计中,我设计的是一个酒店管理系统。通过一学期的学习,对数据库有了一定的认识。在此次的课程设计的过程中,我对具体的设计步骤、思路、方法都有了更深刻的了解,并感受深刻。这次设计使我对数据库中建立基本的表、存储过程、触发器等等都更加熟悉了。但在设计的过程中,由于缺乏经验,某些知识掌握的不全面,还是遇到了许多问题,希望在今后可以完善解决这些问题。

通过这次课程设计发现这其中需要很多知识我们没有接触过,去图书馆查资料时候发现我们前边所学到仅仅是皮毛,还有很多需要我们掌握东西我们根本不知道。同时也发现有很多已经学过东西我们没有理解到位,不能灵活运用于实际,不能很好用来解决问题,这就需要我们不断大量实践,通过不断自学,不断地发现问题,思考问题,进而解决问题。在这个过程中我们将深刻理解所学知识,同时也可以学到不少很实用东西。从各种文档阅读到开始需求分析、概念结构设计、逻辑结构设计、物理结构设计。亲身体验了一回系统设计开发过程。很多东西书上写的很清楚,貌似看着也很简单,思路非常清晰。但真正需要自己想办法去设计一个系统时候才发现其中难度。经常做到后面突然就发现自己开始设计有问题,然后又回去翻工,在各种反复中不断完善自己想法。

总而言之,这次课程设计带给了我很多的收获,希望在接下来的日子里继续加油。

参考文献

- [1] 嵩天等. Python 语言程序设计(第 2 版) [M]. 高等教育出版社, 2017, 2.
- [2] 龚沛曾, 杨志强. Python 程序设计及应用 [M]. 高等教育出版社, 2021, 10.
- [3] 明日科技, 零基础学 Python [M], 吉林大学出版社, 2018, 4.
- [4] 赵广辉等. Python 程序设计基础 [M]. 高等教育出版社, 2021, 5.
- [5] A. silberschatz, H. F. Korth, S. Sudarshan. Database System Concepts (原书第 6 版) [M], 机械工业出版社, 2012, 5.
- [6] 王珊, 张俊. 数据库系统概论 (第 5 版) 习题解析与实验指导 [M]. 高等教育出版社, 2015, 7.
- [7] 胡林玲. 《软件工程与 UML》. 电子工业出版社. 2005.