长 沙 学 院

软件工程基础实训Ⅱ 实训

|  |  |
| --- | --- |
| **题目** | **基于C/S架构的餐饮管理系统** |
| **学院** | **计算机科学与工程** |
| **专业(班级)** | **软件工程（21软件01）** |
| **姓名** | **梁阳** |
| **学号** | **B20210304110** |
| **指导教师** | **杨刚、潘怡** |
| **起止日期** | **2023.06.05～2023.06.16** |

实训任务书

**课程名称：软件工程基础实训II**

**实训题目：基于C/S架构的餐饮管理系统**

### 已知技术参数和设计要求：

1. 问题描述（功能要求）：

本次案例将完成一个基于C/S架构的餐饮管理系统。

（1）.进行餐饮管理系统的功能设计。

（2）.设计餐饮管理系统的数据库。

（3）.利用JAVA的GUI类设计开发友好的人机接口。

（4）.利用JAVA语言设计监听用户操作和JDBC进行数据库的处理。

餐饮管理系统主要有6大模块：

（1）.员工管理：对员工实现增删改查。

（2）.客户管理：对客户实现增删改查。

（3）.餐台管理：对餐台实现增删改查。

（4）.菜品管理：对菜品分类、菜品实现增删改查。

（5）.点菜管理：服务员对某客户、某一空闲餐台实行开台，同时实现点菜，将餐台号与所点的菜品对应起来，分别显示出来，并记录开台时间。

（6）.结账管理：收银员对某一餐台通过统计显示消费的菜品清单统计出消费金额，通过手动输入实收金额进行找零的计算，并显示，完成结账的操作，并记录成统计数据。

基础实训II有2周时间40学时，学生基本必做任务：独立完成基于C/S架构的餐饮管理系统。

选做扩展：菜品推荐、销售统计。

**2. 运行环境要求：**

（1）客户端：

windows操作系统，JDK1.8，Idea。

（2）服务器：

MySQL5.6

3. 技术要求：

1） 掌握软件工程的需求分析和系统设计方法。

2) 掌握数据库的分析与设计，完成餐饮管理系统数据库的设计。

3) 掌握JAVA的GUI设计方法，完成登录、主界面、菜品分类管理界面、菜品管理界面等人机接口。

4） 掌握JAVA面向对象程序设计方法，设计系统所需的各种类。

5) 掌握JDBC方法，对数据库进行操作，完成主程序和菜品分类、菜品管理等功能。

### 实训工作量：

40课时

### 工作计划：

1.班级

21软件1-4

2.课时及教室安排

见实训课程工作计划表。

实训报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 梁阳 | 学号 | B20210304110 | 班级 | 21软件01 |
| 专业 | 软件工程 | | 指导教师姓名 | 杨刚、潘怡 | |
| 项目名称 | 基于C/S架构的餐饮管理系统 | | | | |
| **实训过程**：（说明实训各阶段的工作过程，包括需求分析、系统设计、编码实现、系统测试、交付实施等阶段的主要工作。）  1. 需求分析阶段：  假设我们正在设计一款餐饮管理系统以便于餐厅进行订单管理、库存管理和财务管理。我们可以通过一下步骤完成需求分析：  - 定义用户故事：定义一系列用户故事以便于更好地了解用户的需求，例如“作为餐厅老板，我希望能够查看订单，以便于确定追踪销售量和预测菜品需求”。  - 制定用例：定义餐饮管理系统的用例，例如接单、点餐、、买单等。到最后，我们需要形成一份需求文档，在这个文档中，我们将用例、流程图和用户故事等详细记录下来。  2. 系统设计阶段：  在这个阶段，我们根据需求分析的结果进行系统设计。通过文档，我们了解到系统需要实现什么功能并且我们需要在用户界面上如何呈现这些功能。同时，我们也需要考虑到系统设计的实用性和可扩展性。接下来我们需要做以下工作：  - 数据库设计：针对我们的需求文档中的用例及信息，我们设计合理的数据库并确定关系型及非关系型数据库，包括主键、外键、索引。  - 界面设计：我们将用例及流程图转化为具体的界面呈现方式，包括前台及后台操作界面的设计。  - 业务模块划分：在此阶段我们需要将系统分解成小模块，比如用户管理、订单管理、库存管理等。对于每个模块，我们需要根据用例和流程图进行设计并明确接口。  3. 编码实现阶段：  在确定了系统设计后，就可以开始开发过程了。这个阶段需要开发团队遵守设计规范、编码规范及测试规范，编写易于维护和可扩展的代码。具体工作：  - 定义编码约定：对于命名、包组织、异常机制、类声明、方法声明、注释等编码规范进行约定。团队成员需要按照相同的标准进行开发。  - 小模块开发：根据系统设计划分的模块，进行开发。包括用户管理、订单管理、库存管理等模块的开发，主要涉及接口开发、业务逻辑开发、数据持久化实现等。  - 单元测试：每个模块需要进行单元测试，模块开发完成后开发者必须编写测试类，确保模块功能与预期相符。  4. 系统测试阶段：  在编码实现完成后，需要对系统进行测试。测试阶段需要对系统功能进行全面性和正确性测试，以保证系统的质量。  交互测试：我们需要确保系统的界面与数据库能有效交互。通过交互测试，我们可以确保用户可以顺利访问系统并进行各种操作。 | | | | | |
| **实训成果**：  参见基于C/S架构的餐饮管理系统设计说明书。 | | | | | |
| **实训总结**：（包括心得体会、存在的问题和改进方向。）  1. 设计心得  - 数据库架构设计要合理：在开发系统时，数据库的设计是至关重要的一个环节。一个合理的数据库设计能够提高系统性能和可维护性。在实际开发中，我们应该遵循约束完整性、范式化设计等规范，减少冗余的数据存储，提高查询效率。  - 泛型方法可以使方法的输入参数更加灵活性，也可以增强程序的类型检查，避免了一些不必要的类型转换工作，增加了程序的安全性和稳定性。同时，泛型方法也方便了程序员的编程，可以减少代码的重复性和提高代码的可读性。  - 模块划分要清晰：在设计系统时，我们需要将系统划分成多个独立模块，以便于系统的拓展和维护。不同的模块负责不同的功能，通过接口相互交互，提高系统的整体性能和可维护性。  2. 存在的问题  - 用户体验不够友好：在系统用户使用过程中，用户体验是至关重要的，但是在实际开发中，往往忽略了用户体验。系统的UI界面设计可以更具人性化，简洁明了，有助于用户操作。  - 安全性问题：餐饮管理系统涉及许多重要的用户数据、订单数据、财务数据等，系统的数据安全问题是用户非常关心的问题。在实际开发中，需要加强对系统安全性的保护，确保敏感数据不会被泄露。  - 功能不够完善：虽然我们在设计时考虑了用户需求，但是在实际应用中，用户对餐饮管理系统的要求可能还需要进一步补充。  3. 改进方向：  - 提高用户体验：设计更符合人性化的UI界面，方便用户进行操作、遵循常用的按钮布局等，来提高用户体验。  - 数据安全保护：加强数据加密，防止SQL注入等安全问题的出现，防止未经授权的访问。  - 拓展新的功能：不断与餐饮业者沟通，开发更多新的功能来满足他们的需求，提高系统的完整性和适应性。  - 加强系统的整体性能：在开发系统时，不仅需要考虑到单个模块的性能优化，还需要考虑到整个系统的综合性能优化，让系统的运行更加流畅和高效。 | | | | | |
|  | | | | | |

基于C/S架构的餐饮管理系统

设计说明书

作者：张伟(B20210304225)

计算机科学与工程学院

2023年 06 月 15日

**摘要**

本课程设计基于C/S架构开发了一款餐饮管理系统，该系统具备订单管理、菜品管理、员工管理、结账管理等常用功能。在设计中，我们注重数据库架构的合理设计、技术选型的合适性、模块划分的清晰性等方面的考虑。通过不断地与餐饮业者对接，我们不断地拓展新的功能，同时不断地加强系统的安全保护和整体性能优化。

使用Java中的泛型方法进行编程，提高了程序的灵活性和安全性。经过测试，该系统能够满足餐饮业者增长的需求，提高了他们的管理效率，获得了很好的实际效果。

在模块划分方面，我们注重了模块之间的独立性和通用性。例如，在订单管理模块中，我们设计了订单的添加、查询、修改、删除等操作，同时还提供了订单状态的跟踪和统计分析的功能，方便餐饮业者监督订单状态和进行数据分析。

在设计系统的数据库结构时，我们注重了数据的规范性和完整性，通过建立订单详情表、菜品表等，使数据在系统中能够得到完整和准确的处理，实现了数据的有效性和一致性。

1. 订单管理：用户可以通过前台进行下单、支付、查询、查看订单详情等操作，商家可以通过后台进行订单的处理、跟踪、统计等操作。同时，商家还可以根据订单状态对订单进行管理和处理。
2. 菜品管理：商家可以通过后台进行菜品的添加、删除、修改、查询以及上传菜品图片等操作。同时，商家还可以对菜品进行分类和标签的管理，以便用户进行查找和筛选。
3. 桌台管理：商家可以通过后台对餐厅的桌台进行管理和分配，可以查看桌台当前状态、开台信息、点菜信息等。同时，商家还可以设置桌台容纳人数、规定就餐时间等相关参数，以优化餐厅接待服务和用餐体验。
4. 结账管理：商家可以通过后台进行财务管理、收支记录和费用统计等操作，以进行财务分析和资金管理。商家还可以对财务数据进行导出和存档以便进行更深层次的数据分析和决策。

餐饮管理系统采用了C/S架构进行开发，客户端采用了Java Swing技术进行界面设计和开发，服务器端使用Java编程语言进行开发。

系统的后端开发采用了JDBC技术进行数据访问和操作，同时还使用了MySQL关系型数据库进行数据存储和管理。在业务功能方面，系统实现了订单管理、菜品管理、桌台管理和财务管理等功能，并提供了一些数据统计和报表生成的功能。

系统在安全方面实现了登录认证和权限管理等功能，确保了商家和用户的数据安全。在性能优化方面，系统进行了数据库的优化和连接池的配置，在数据存取方面表现优异。

总之，餐饮管理系统采用C/S架构，实现了订单管理、菜品管理、桌台管理和财务管理等功能，具有数据统计和报表生成功能，采用Java Swing等技术进行客户端开发，使用Java编程语言进行服务器端开发，使用JDBC进行数据操作，并使用MySQL进行数据存储，并保证了数据安全和性能优异。

目录

[已知技术参数和设计要求： 2](#_Toc138103902)

[1. 问题描述（功能要求）： 2](#_Toc138103903)

[**2. 运行环境要求：** 2](#_Toc138103904)

[3. 技术要求： 2](#_Toc138103905)

[实训工作量： 3](#_Toc138103906)

[工作计划： 3](#_Toc138103907)

[1 引言 1](#_Toc138103908)

[1.1 编写目的 1](#_Toc138103909)

[1.2 参考资料 4](#_Toc138103910)

[2 需求规约 6](#_Toc138103911)

[2.1 功能需求 6](#_Toc138103912)

[2.2 界面需求 9](#_Toc138103913)

[2.3 数据需求 9](#_Toc138103914)

[3 系统设计 10](#_Toc138103915)

[3.1 运行环境 11](#_Toc138103916)

[3.2 系统静态结构设计 11](#_Toc138103917)

[3.3 人机接口设计 13](#_Toc138103918)

[3.4 结账模块设计说明 15](#_Toc138103919)

[3.5 点菜模块设计说明 17](#_Toc138103920)

[4 数据库设计 20](#_Toc138103921)

[4.1 数据库环境说明 20](#_Toc138103922)

[4.2 数据库的命名规则 20](#_Toc138103923)

[4.3 逻辑结构设计 20](#_Toc138103924)

[4.4 物理结构设计 21](#_Toc138103925)

[4.5 安全性设计 22](#_Toc138103926)

[5 测试用例设计 24](#_Toc138103927)

[附录 26](#_Toc138103928)

[附录A 本系统的组成 26](#_Toc138103929)

[附录B 程序源代码(经过注释的关键代码) 26](#_Toc138103930)

# 1 引言

## 1.1 编写目的

一、引言

本文档旨在明确“C/S架构来开发一款餐饮管理系统”软件的需求规格、概要设计、详细设计和功能测试用例等内容，以提供给需求分析人员、系统设计人员、系统开发人员、系统测试人员等相关人员作为软件开发及维护的指导。

二、软件需求

1. 功能性需求

（1）订单管理：实现用户下单、支付、查询，商家对订单的管理和处理等功能。

（2）菜品管理：实现商家添加、删除、修改、查询菜品，以及菜品分类和标签的管理等功能。

（3）桌台管理：实现商家对餐厅桌台的管理和分配功能，显示桌台状态，设置桌台容纳人数、规定就餐时间等相关信息。

（4）财务管理：实现商家进行账务管理、收支明细和费用统计等功能，进行财务分析和资金管理。

（5）点菜管理：顾客可以通过系统浏览菜品列表，选择喜欢的菜品进行点菜，还可以按需求修改菜品属性等信息。主要用户需求是方便快捷的点菜操作，让顾客能够更好地享受就餐体验。

2. 非功能性需求

（1）安全性：确保商家和用户的私密数据不被非法获取或篡改。

（2）可靠性：整个系统具有较好的可靠性和稳定性，保证在耐受性范围内的故障停机时间尽可能少。

（3）性能：系统具有良好的性能表现，具备较好的响应速度和并发处理能力。

三、概要设计

1. 模块划分

（1）客户端模块：采用Java Swing技术进行界面设计和开发，实现用户操作界面。

（2）服务器模块：采用Java编程语言进行开发，使用Socket通信进行客户端和服务器之间的交互。

2. 架构设计

（1）C/S架构：采用客户端/服务器架构，实现前后端分离。

（2）JDBC操作：在服务器端使用JDBC进行数据访问和操作。

（3）MySQL数据库：使用MySQL关系型数据库进行数据存储和管理。

四、详细设计

1. 数据库设计

（1）订单表：包含订单号、用户ID、菜品ID、桌台ID、订单金额、下单时间、支付时间、订单状态等字段。

（2）菜品表：包含菜品ID、菜品名称、菜品价格、菜品描述、菜品图片等字段。

（3）桌台表：包含桌台ID、桌台名称、桌台容纳人数、桌台状态等字段。

2. 应用程序设计

（1）客户端：采用Java Swing技术进行界面设计和开发，主要模块包括订单管理、菜品管理、桌台管理、财务管理等。

（2）服务器端：采用Java编程语言进行开发，主要模块包括订单模块、菜品模块、桌台模块、财务模块等。

五、功能测试用例

1. 订单管理模块测试用例

（1）测试项目：下单

（2）测试操作：用户点击菜品，将菜品加入购物车，确认后下单。

（3）预期结果：订单状态更新，成功下单。

2. 菜品管理模块测试用例

（1）测试项目：菜品添加

（2）测试操作：商家在菜品管理界面点击添加，填写菜品信息并上传图片。

（3）预期结果：菜品信息成功添加到数据库，图片上传成功。

3. 桌台管理模块测试用例

（1）测试项目：桌台分配

（2）测试操作：商家在桌台管理界面进行桌台分配。

（3）预期结果：桌台状态更新，分配成功。

4. 结账管理模块测试用例

（1）测试项目：收支统计

（2）测试操作：商家在财务管理界面查看每日收支统计数据。

（3）预期结果：显示每日收入、支出等统计数据。

## 1.2 参考资料

以下是本文档中参考资料的详细信息：

1. Java编程思想 第四版

作者：Bruce Eckel

版本：第四版

出版社：机械工业出版社

2. Java Swing实战 第二版

作者：沈健民

版本：第二版

出版社：人民邮电出版社

3. Java核心技术卷II 高级特性

作者：雷卫民

版本：第10版

出版社：机械工业出版社

4. Spring实战 第二版

作者：Craig Walls

版本：第二版

出版社：人民邮电出版社

5. MyBatis技术内幕 深入解析MyBatis核心原理与最佳实践

作者：王松

版本：第二版

出版社：电子工业出版社

6. 论文：基于Java Swing的快餐店前台管理系统设计与实现

作者：李慕阳，王建旋

期刊：计算机与现代化

卷期：2019年第07期

7. 网络资源：Java菜鸟教程 - Java Swing

作者：菜鸟教程

网址：https://www.runoob.com/w3cnote/java-swing-tutorial.html

# 2 需求规约

## 2.1 功能需求

程序的功能需求主要包括订单管理、菜品管理、桌台管理、财务管理等几个方面，下面对每个功能进行详细说明：

1. 订单管理：用户和商家都可以对订单进行管理，实现用户下单、支付、查询，商家对订单的管理和处理等功能。主要用户需求是实现快速、方便、准确的订单管理和处理。

2. 菜品管理：商家可以对菜品进行添加、删除、修改、查询，以及菜品分类和标签的管理等功能。主要用户需求是方便易用的菜品管理，有助于提高效率和服务质量。

3. 桌台管理：商家可以对餐厅桌台进行管理和分配功能，显示桌台状态，设置桌台容纳人数、规定就餐时间等相关信息。主要用户需求是方便餐厅桌台管理，提供给顾客更好的服务体验。

4. 结账管理：商家可以进行账务管理、收支明细和费用统计等功能，进行财务分析和资金管理。主要用户需求是方便、准确、稳定的财务管理，有助于提高餐饮经营效率。

1. 点菜管理：顾客可以通过系统浏览菜品列表，选择喜欢的菜品进行点菜，还可以按需求修改菜品属性等信息。主要用户需求是方便快捷的点菜操作，让顾客能够更好地享受就餐体验。

2.1.1 点菜管理用例说明

以下基于用户角度，对点菜管理的使用过程进行用例说明，包括输入和输出：

1. 顾客点菜用例：

输入：

- 用户打开系统点餐页面，浏览菜品列表，选择喜欢的菜品；

- 用户可以按需求修改菜品属性信息，如份量、口味等；

- 用户确认订单后，将订单提交给商家后台管理。

输出：

- 用户得到一份确认订单单据，包括选菜内容、价格、订单数量等信息；

- 商家得到订单信息后，根据订单要求进行烹饪制作，准备配送到相关桌台。

异常处理

1. 支付处理异常：如果用户选择了在线支付，并支付出现问题，应给出友好的提示信息，提示用户重新支付或联系客服解决问题。
2. 菜品信息异常：如果系统中菜品信息有误，如价格设置错误等，应进行及时纠正，并提供相应的处理措施，如退款或补差价等。

2.1.2 结账管理用例说明

1. 顾客结账用例：

输入：

- 用户完成用餐后，需要进行结账操作；

- 用户打开结账页面，输入订单号，查询订单相关信息；

- 用户确定订单信息后，选择支付方式进行付款。

输出：

- 用户得到一份确认订单单据，包括选菜内容、价格、订单数量等信息；

- 商家得到订单信息后，根据订单要求进行烹饪制作，准备配送到相关桌台。

活动步骤：

- 用户打开结账页面，输入订单号

- 系统查询订单相关信息后，显示订单商品信息、总价格；

- 用户选择支付方式后，系统生成支付二维码或跳转支付页面；

- 用户完成支付后，系统显示订单支付状态，并生成订单结账单据。

扩展点：

- 第3步中，如果系统暂不支持用户所选择的支付方式，系统应该通过友好提示让用户选择其他支付方式；

- 第4步中，如果用户选择了线下支付，需要调用系统的桌台管理功能，对桌台进行状态更改，同时向厨房发送订单通知。

异常处理：

- 如果在第 1 步中，用户输入订单信息时出现错误，系统应给出友好提示，提示用户检查订单信息是否正确；

- 如果在第 3 步中，支付系统出现异常，系统应给出相应的提示信息，并提醒用户稍后再尝试支付，或提示用户选择其他支付方式；

- 如果在第 4 步中，订单提交到后台或者打印账单时出现错误，系统应该给出相应错误提示，提醒用户重新提交订单。

2. 商家账务管理用例：

输入：

- 商家使用系统中的账务管理功能，实现对财务收支、费用统计、利润分析等功能的管理；

- 商家可以查看各项数据指标，如交易总额、消费结构、运营利润等，并进行数据的图表分析。

输出：

- 商家能够及时了解到餐厅财务状况，以便及时制定相应的经营管理策略和政策，提高经营效率和效益。

活动步骤：

- 商家登录系统的账户后，进入餐饮管理中心；

- 商家可以选择查看总体财务情况，包括总收入、总支出、总利润等，还可以查看各项数据指标，如交易总额、消费结构、品类分析、市场份额等；

- 商家还可以查看单个订单、用户消费记录、交易流水等详细数据，如实现退款、收据打印等功能。

扩展点：

- 在第 2 步中，商家可以选择查看各项指标的特定时间区间，并设置时间范围和统计方式；

- 商家可以设置财务收支警戒线，提醒自己及时调整经营策略和费用管理方式。

异常处理：

- 如果在第 1 步中，商家无法正常登录系统，系统应给出友好的提示信息，帮助商家解决登录问题；

- 如果在第 2 步中，系统出现错误或数据异常，应给出相应的提示信息，并提醒商家确认数据或重新查询。

以上基于用户角度，对结账管理系统的输入和输出以及可能出现的异常进行了简要说明，希望对理解该系统的运行原理和过程有所帮助。

## 2.2 界面需求

1. 易用性：交互界面应该简单易用，让用户能够轻松地理解和操作系统，提高用户的使用效率和体验。
2. 直观性：交互界面应该能够呈现清晰、明确的信息，让用户能够直观地了解系统的功能和操作方式，避免用户在使用过程中出现疑惑或迷惑。
3. 美观性：交互界面应该呈现美观、舒适的外观设计，让用户感觉愉悦并愿意不断使用系统。良好的界面设计还可以提高系统的信任度和认可度。
4. 交互响应性：交互界面应该保持快速和灵敏的响应速度，对用户操作做出及时的反应，避免用户对操作产生不良体验和情绪。
5. 模块化和可扩展性：交互界面应该基于清晰的架构设计，保持模块化和可扩展性的特性，与系统中其他模块能够进行良好的协同和集成。

## 2.3 数据需求

在点菜和结账管理系统中，程序需要处理的数据实体包括：

1. 用户实体。

属性：

- 用户名（username）: 字符串类型，用于标识用户唯一身份，长度 1-50，不允许为空，作为唯一标识符；

- 密码（password）: 字符串类型，用于用户身份验证，长度 6-16，不允许为空，需要加密存储；

- 手机号（phone）: 字符串类型，用户联系方式，长度 11，不允许为空，需要进行格式检查，并保证唯一性；

- 邮箱（email）: 字符串类型，用户联系方式，长度 50，允许为空，需要进行格式检查，并保证唯一性；

- 姓名（name）: 字符串类型，用户真实姓名，长度 2-30，不允许为空；

- 地址（address）: 字符串类型，用户住址，长度 10-100，可以为空。

数据存储要求：

- 用户信息需要进行安全存储，密码需要进行加密处理，并采用安全的存储方式，防止泄露。

数据处理过程：

- 用户信息进行注册、登录、修改、查询时，需要进行相关的数据验证和加密处理；

- 验证用户身份时，需要对输入的用户名和密码进行校验；

- 用户信息修改和查询时，需要对数据进行及时更新和有效查询；

- 用户信息删除时，需要进行安全的数据清除处理。

表2.1 用户实体



# 3 系统设计

图2.1 图书管理系统功能模块图

以数据为核心划分模块。（不是以角色划分模块）

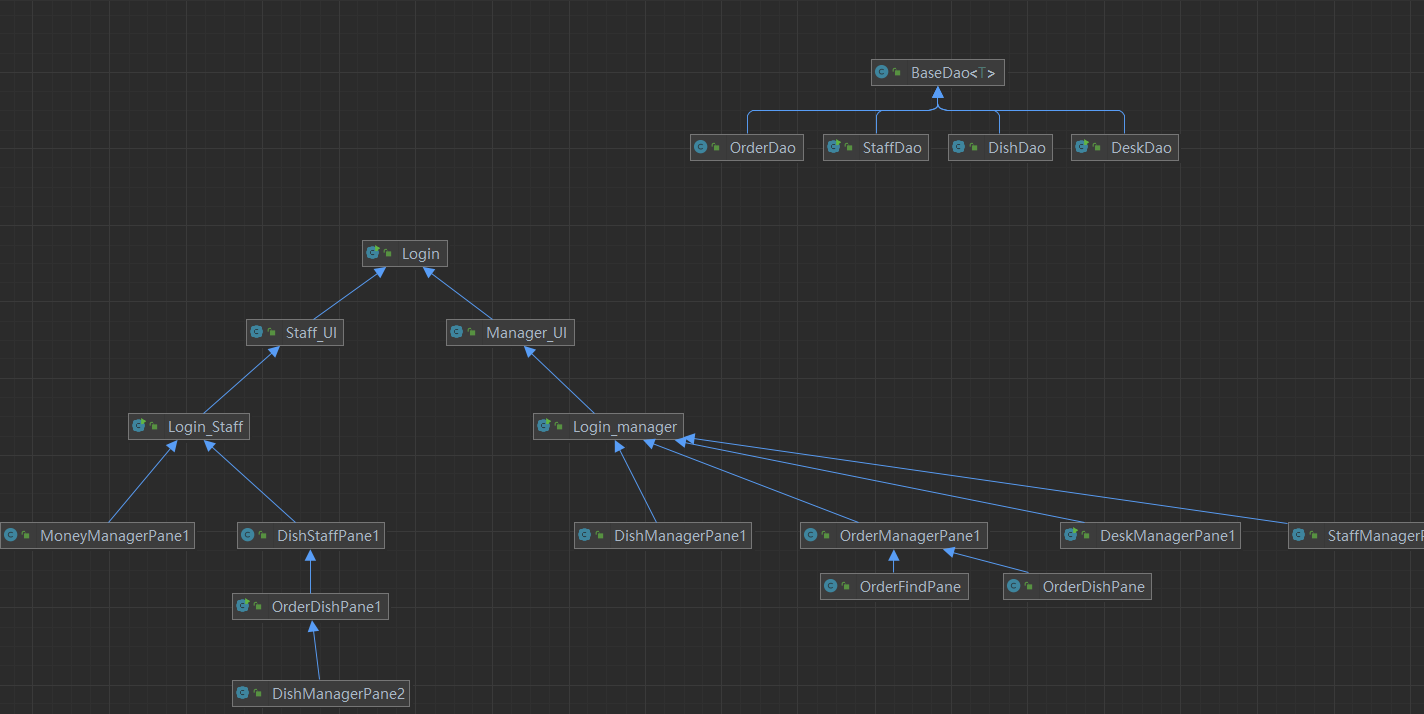
## 3.1 运行环境

软件环境：JRE8

## 3.2 系统静态结构设计

画出每个功能模块对应的的视图层（View，界面）、服务层（Service,业务）和模型层（Model,实体层）。

假设此系统的组成如下：（请完善下图，掉其中的。。。）



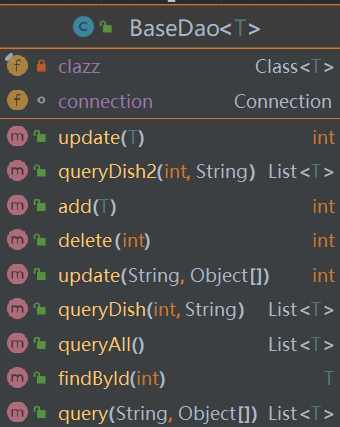


图3.1 某个模块服务类图

系统静态结构说明

在上述实现中，泛型的设计体现了功能需求和非功能需求的考虑。

首先，我们的功能需求是实现数据库的动态增删改查等功能，因此需要定义一个抽象的`BaseDao`接口，来定义增删改查的方法签名。这个接口中使用了泛型`<T>`，以便在实现时可以动态地传递类型参数以适合各种数据类型的实体类。这符合了泛型设计的基本原则之一：使代码更具有可重用性和泛化性。

其次，我们需要对一些非功能需求进行考虑。例如，我们需要保证程序的性能、安全性和可维护性，因此在具体实现的时候，我们需要使用反射技术实现动态构造 SQL 语句，以避免手写大量 SQL 代码所导致的错误和低效率。这也充分体现了泛型设计的另一个原则：尽可能地抽象和泛化，实现共性和复用性，从而保证代码的可维护性和可扩展性。

还有，我们需要对代码的可读性和可维护性进行优化，在 DAO 类中使用了 PreparedStatement 对象和 try-with-resources 语句，以更好地控制连接和资源的释放，同时避免了 SQL 注入等安全风险。这也体现了良好的编程实践：遵循高质量的编程规范和标准，提高代码的可读性、可维护性和安全性。。

## 3.3 人机接口设计

3.3.2结账窗体1设计

说明本接口的界面设计，给出界面布局。例如：图3.1

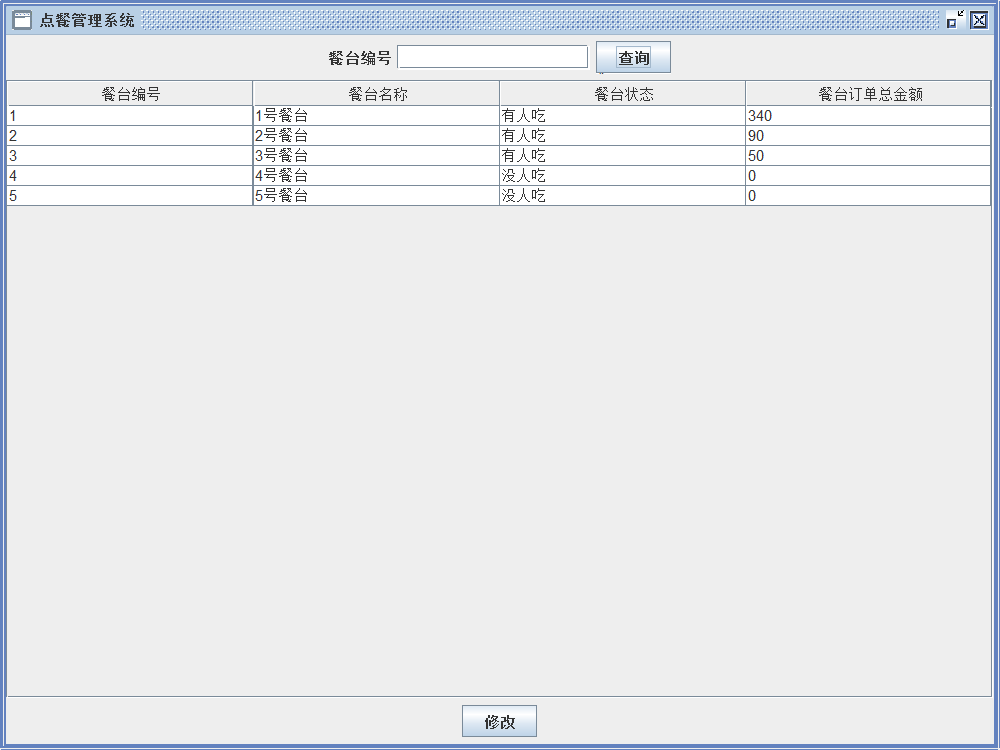


图3.2 新增入库单界面布局设计

界面元素主要控件：

1. JFrame：用于创建主窗体。

2. JPanel：用于创建面板，将控件组装在面板上。

3. JLabel：用于创建文本标签。

4. JTextField：用于创建单行文本输入框。

5. JButton：用于创建按钮。

6. JDialog：用于创建对话框。

7. JTable：用于创建表格。

8. DefaultTableModel：用于设置表格的数据模型。

9. JScrollPane：用于创建滚动面板，将表格放在滚动面板中，支持表格的滚动浏览。

10. Box：用于创建容器，用于存放其他控件。

11. Dimension：用于设置控件的尺寸。

12. Toolkit：用于获取屏幕尺寸。

13. BorderLayout：用于设置窗体的布局方式。

14. FlowLayout：用于设置面板中控件的排列方式。

15. ActionListener：用于处理按钮点击事件。

功能说明：

1. 查询功能：通过输入餐台编号，点击查询按钮，可以查询到对应餐台的订单信息。

2. 表格显示：将查询结果在表格中展示，包括餐台编号、餐台名称、餐台状态和订单总金额。

3. 添加菜品功能：点击添加菜品按钮，弹出对话框，输入菜品信息，确认后将菜品添加到菜品列表中。

4. 删除功能：选中表格中的餐台记录，点击删除按钮，将该餐台从数据库中删除。

5. 修改功能：选中表格中的餐台记录，点击修改按钮，弹出对话框，可以修改餐台信息。

6. 更新表格：在对餐台进行添加、删除、修改等操作后，可以点击更新按钮，将最新的餐台信息显示在表格中。

## 3.4 结账模块设计说明

功能模块1：餐台管理，包括餐台的增删改查

视图层：

- MoneyManagerPane1：主界面，包含查询餐台、展示餐台列表的表格和按钮面板。

- JDialog：用于添加餐台和修改餐台的对话框。

服务层：

- DeskDao：餐台数据访问对象，用于与数据库交互，包含增删改查餐台的方法。

模型层：

- Desk：餐台实体类，包含餐台编号、餐台名称、餐台状态等属性。

功能说明：

1. 查询餐台功能：在界面的查询框中输入餐台编号，点击查询按钮，调用DeskDao的查询方法，返回符合条件的餐台列表，在表格中展示出来。

2. 展示餐台列表功能：使用JTable和DefaultTableModel来展示餐台列表，表格的每一行对应一个餐台信息。

3. 添加餐台功能：点击添加菜品按钮，弹出添加餐台的对话框，在对话框中输入餐台信息，点击确认按钮后调用DeskDao的添加方法，将餐台信息添加到数据库中，并更新表格显示。

4. 删除餐台功能：选中表格中的一行餐台记录，点击删除按钮，调用DeskDao的删除方法，将该餐台从数据库中删除，并更新表格显示。

5. 修改餐台功能：选中表格中的一行餐台记录，点击修改按钮，弹出修改餐台的对话框，在对话框中修改餐台信息，点击确认按钮后调用DeskDao的修改方法，更新餐台信息，并更新表格显示。

6. 更新表格功能：在对餐台进行添加、删除、修改等操作后，点击更新按钮，调用DeskDao的查询方法，获取最新的餐台列表，并更新表格显示。

类说明：

- MoneyManagerPane1类是主界面类，包含餐台的查询和展示功能，以及添加、删除、修改餐台的按钮监听事件。

- DeskDao类是餐台数据访问对象，提供对数据库中餐台表的增删改查操作的具体实现。

- Desk类是餐台实体类，定义餐台的属性，包括餐台编号、餐台名称、餐台状态等。

方法说明：

- MoneyManagerPane1类：init()方法用于初始化界面及各个控件，updateTable()方法用于更新表格显示。

- DeskDao类：queryAll1()方法用于查询所有餐台信息，findById1()方法用于根据餐台编号查询餐台信息，update()方法用于执行更新数据库的操作。

- Desk类：getters和setters方法用于设置和获取餐台的属性值。

3.4.1 设计应用类

根据功能模块的划分，可以确定以下边界类：

- MoneyManagerPane1：主界面类，通过用户接口与用户交互，包括查询、展示、添加、删除、修改餐台的功能。

- JDialog：对话框类，用于添加和修改餐台信息。

控制类和持久类的设计可以如下所示：

- 控制类：

- DeskController：负责处理餐台相关的业务逻辑，包括查询、添加、删除、修改餐台等操作。与界面类进行交互，调用相关的持久类来完成数据的存取。

- 持久类：

- DeskDao：数据访问对象，负责与数据库进行交互，提供对餐台表的增删改查操作。

- DishDao：数据访问对象，负责与数据库进行交互，提供对菜品表的增删改查操作。

应用类模型中的关系如下：

- MoneyManagerPane1与DeskController：通过MoneyManagerPane1的事件监听器，调用DeskController中的方法来处理业务逻辑。

- DeskController与DeskDao：DeskController通过调用DeskDao的方法来进行数据的存取和操作。

- DeskController与MoneyManagerPane1：DeskController将处理结果返回给MoneyManagerPane1，MoneyManagerPane1根据结果更新界面显示。

- DeskDao与数据库：DeskDao与数据库进行交互，执行SQL语句来实现对餐台的增删改查操作。

整体的关系为：MoneyManagerPane1（界面层）-->DeskController（控制层）-->DeskDao（持久层）-->数据库

## 3.5 点菜模块设计说明



设计考虑如下：

1. 根据功能需求，设计了一个用于显示菜品信息的界面类OrderDishPane1。

2. 通过OrderDishPane1类的构造方法，传入餐台ID来初始化界面。

3. 在界面中使用JTable来显示菜品信息，使用DefaultTableModel作为表格的数据模型。

4. 通过DishDao类从数据库中获取菜品信息，并将菜品信息填充到表格中。

5. 为了方便操作，添加了一个按钮用于添加菜品。

6. 点击添加菜品按钮会弹出一个确认对话框，提示是否确认增加该菜品到订单中。

7. 如果确认增加菜品，则调用OrderDao的添加方法，将菜品信息和订单信息插入到数据库中。

8. 更新表格的显示，通过调用updateTable()方法重新从数据库中获取最新的菜品信息。

9. 为了美观，对表格进行了样式和布局的调整，使得菜品信息居中显示。

设计考虑主要围绕功能需求展开，使用了适当的UI组件和数据模型，同时考虑到用户交互和界面的美观性，使得用户能够方便地操作和浏览菜品信息。

功能模块：菜品管理

视图层：

- OrderDishPane1：主界面类，显示菜品信息和添加菜品功能。

服务层：

- DishDao：菜品数据访问对象，用于与数据库交互，提供菜品的增删改查操作。

模型层：

- Dish：菜品实体类，包含菜品的编号、名称、类别、价格、状态等属性。

功能说明：

1. 显示菜品信息功能：通过界面类OrderDishPane1将菜品信息显示在表格中，通过调用DishDao的查询方法获取菜品信息，通过调用createTableModelData方法将菜品信息转换为表格需要的数据。

2. 添加菜品功能：在界面中点击添加菜品按钮，弹出确认对话框，并调用DishDao的添加方法将菜品信息插入到数据库中。

3. 更新表格功能：在进行添加菜品操作后，调用updateTable方法重新从数据库获取最新的菜品信息，并更新表格显示。

类说明：

- OrderDishPane1类是主界面类，用于显示菜品信息和添加菜品功能，通过调用DishDao的方法和更新表格方法来完成相关操作。

- DishDao类是菜品数据访问对象，提供对数据库中菜品表的增删改查操作，与数据库进行交互。

- Dish类是菜品实体类，包含菜品的属性，用于存储菜品的编号、名称、类别、价格、状态等信息。

方法说明：

- OrderDishPane1类：

- 构造方法OrderDishPane1：传入餐台ID，初始化界面，设置按钮的监听器，通过调用DishDao的方法和更新表格方法来完成相关操作。

- updateTable方法：从数据库获取最新的菜品信息，并更新表格显示。

- DishDao类：

- queryAll1方法：查询所有菜品信息，返回一个List<Dish>对象。

- insert方法：插入一条菜品记录到数据库，接收Dish对象作为参数。

- Dish类：

- getters和setters方法：用于获取和设置菜品的属性值。

每个类中的属性和方法主要用于实现相应的功能，例如DishDao类的queryAll1方法用于查询所有菜品信息，insert方法用于插入一条菜品记录，Dish类的属性用于存储菜品信息。这些方法和属性的具体用途在功能说明中已经提到了。

# 4 数据库设计

## 4.1 数据库环境说明

Navicat目前最新版本是Navicat Premium 15.0，它是一款跨越多种数据库管理系统的工具，支持MySQL、MariaDB、Oracle、SQLite、PostgreSQL 和 SQL Server等数据库，同时提供了多种实用工具和特性，例如跨库查询、数据同步、数据备份和还原等功能。

## 4.2 数据库的命名规则

数据库命名规则是一个重要的规范，它可以让数据库设计更加清晰、易于维护。下面是数据库命名规则的一些说明：

1. 数据库名的格式

数据库名应该由字母、数字、下划线、中划线等符号组成，不能包含特殊字符，如空格、斜杠、反斜杠、问号等。数据库名一般使用小写字母，用下划线或中划线分隔单词，以此提高数据库名的可读性和可维护性。例如：mydatabase、my\_database、my-database 等。

2. 数据表名的格式

数据表名的命名方式一般与数据库名类似，只不过数据表名要尽量简明、易于理解。数据表名应该是复数形式，遵循小写字母、下划线或中划线的命名规则。例如：users、user\_info、user-info 等。

3. 字段名的格式

字段名应该是小写字母，遵循下划线或驼峰式的命名规则，字段名应该具有一定含义，能够简明明确地表示字段所表示的数据。例如：user\_id、user\_name、userAge 等。

4. 约束名的格式

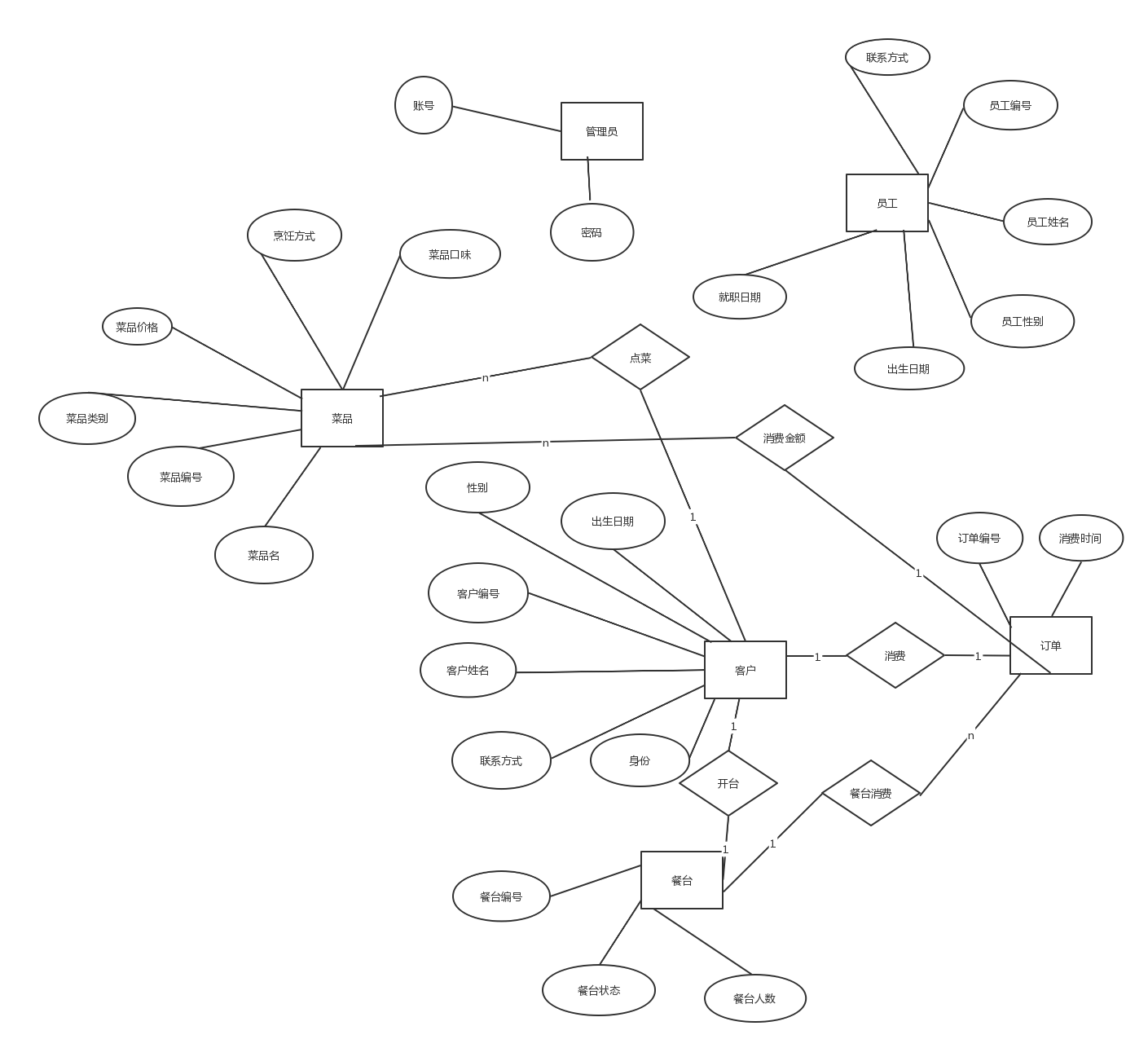
约束名是指用于保证数据完整性和正确性的限制条件，它可以通过命名来标识其所承担的约束类型，如主键、唯一、外键等。约束名一般应该是由表名、字段名、约束类型等组合而成的表达式，例如：pk\_users、uk\_user\_name、fk\_user\_info 等。

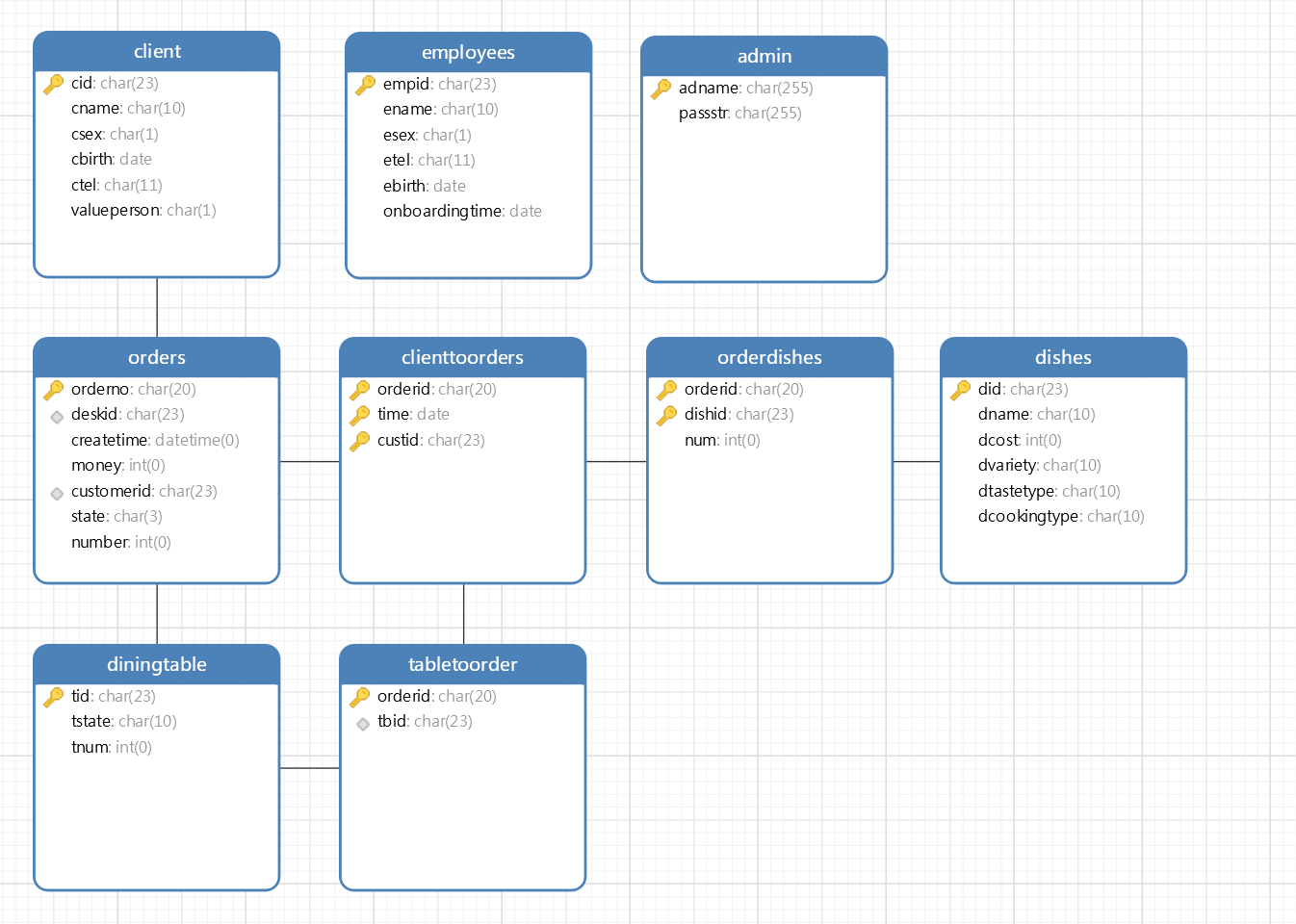
5. 触发器名的格式

触发器是一种特殊的程序，它是一组在表中自动执行的 SQL 语句，用于维护表之间的关系和数据的完整性。触发器的命名应该具有一定的描述性，包括触发时机、触发的表、触发类型、触发操作等信息。例如：on\_users\_insert、on\_user\_info\_update 等。

综上所述，数据库命名规则是一种重要的约定，它可以让数据库设计更加规范、易于维护。好的命名规则应该能够清晰地表达命名对象的含义，提高系统的可读性和可维护性。

## 4.3 逻辑结构设计





## 4.4 物理结构设计

## 4.4.1 数据库表

说明数据库的表的设计。

表4.1 表汇总

|  |  |
| --- | --- |
| 表 名 | 功 能 说 明 |
| customer | 员工管理 |
| Desk | 对餐台进行管理 |
| Dish | 对菜品进行管理 |
| Order | 存所有的订单 |
| OrderDish | 存订单下所有的菜 |
|  |  |
|  |  |

表4.2 用户实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| adname | char | 255 | 否 | 是 | 用户名 |
| passstr | char | 255 | 是 | 否 | 密码 |

表4.3客户实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| cid | char | 23 | 否 | 是 | 客户号 |
| cname | char | 10 | 是 | 否 | 姓名 |
| csex | char | 1 | 是 | 否 | 性别 |
| cbirth | date | 0 | 是 | 否 | 出生日期 |
| ctel | char | 11 | 是 | 否 | 电话号码 |
| valueperson | char | 1 | 是 | 否 | 会员 |

表4.4餐台实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| tid | char | 23 | 否 | 是 | 餐台号 |
| tstate | char | 10 | 是 | 否 | 状态 |
| tnum | int | 0 | 是 | 否 | 座位数 |

表4.5菜品实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| did | char | 23 | 否 | 是 | 菜品号 |
| dname | char | 10 | 是 | 否 | 菜名 |
| dcost | int | 0 | 是 | 否 | 价格 |
| dvariety | char | 10 | 是 | 否 | 品种分类 |
| dtastetype | char | 10 | 是 | 否 | 口味分类 |
| dcooking | char | 10 | 是 | 否 | 烹饪方法 |

表4.6员工实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| empid | char | 23 | 否 | 是 | 员工号 |
| ename | char | 10 | 是 | 否 | 姓名 |
| esex | char | 1 | 是 | 否 | 性别 |
| etel | char | 11 | 是 | 否 | 电话号码 |
| ebirth | date | 0 | 是 | 否 | 出生日期 |
| onboardingtime | date | 0 | 是 | 否 | 入职日期 |

表4.7订单实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| Orderno\_id | char | 20 | 否 | 是 | 订单编号 |
| deskid | char | 23 | 是 | 否 | 餐台号 |
| createtime | datetime | 0 | 是 | 否 | 就餐时间 |
| money | int | 0 | 是 | 否 | 消费金额 |
| customerid | char | 23 | 是 | 否 | 客户编号 |
| state | char | 3 | 是 | 否 | 状态 |
| number | int | 0 | 是 | 否 | 就餐人数 |

表4.8总订单实体

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 属性名称 | 类型 | 长度 | 是否为空 | 唯一标识 | 说明 |
| id | char | 20 | 否 | 是 | 订单编号 |
| deskid | char | 23 | 是 | 否 | 餐台号 |
| createtime | datetime | 0 | 是 | 否 | 就餐时间 |
| money | int | 0 | 是 | 否 | 消费金额 |
| customerid | char | 23 | 是 | 否 | 客户编号 |
| state | char | 3 | 是 | 否 | 状态 |
| number | int | 0 | 是 | 否 | 就餐人数 |

## 4.5 安全性设计

4.5.1 防止用户直接操作数据库的方法

（1）用户标识和鉴定

（2）存取控制

（3）数据加密

4.5.2 用户帐号密码的加密方法

防止用户直接操作数据库的方法。用户只能用帐号登录到应用软件，通过应用软件访问数据库，而没有其他途径操作数据库。用户帐号密码的加密方法。对用户帐号的密码进行加密处理，确保在任何地方都不会出现密码的明文。

# 5 测试用例设计

表5.1 登录-注册测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 登录 | yl-001 | 输入‘用户名’和‘密码’ | 2942990081，1237236182 | 登录成功 | 通过 |
| yl-002 | 2942990081，1234567 | 登录失败 | 通过 |
| yl-003 | 29429900811，1237236182 | 登录失败 | 通过 |
| yl-004 | 2942990081,空 | 登录失败 | 通过 |
| yl-005 | 空,1237236182 | 登录失败 | 通过 |
| yl-006 | 2942990081,一个空格 | 登录失败 | 通过 |
| yl-007 | 一个空格，1237236182 | 登录失败 | 通过 |
| yl-008 | 2942990081，密码长度超过32 | 登录失败 | 通过 |
| yl-009 | 用户名超过36，1237236182 | 登录失败 | 通过 |
| 注册 | yl-010 | 用户名，密码 | 8888888888,23216323 | 创建成功 | 通过 |
| y1-011 | 用户名，密码 | 8888888888,1 | 创建失败 | 通过 |

表5.2 员工管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 增加员工 | y2-001 | 点击添加按  钮，输入‘姓  名’，‘性别’，‘出生日期’，‘电话号码’，‘就职日期’ | Anish，女，15273751972,  2002-08-03,2022-06-18 | 新增成功 | 通过 |
| 删除员工 | y2-002 | 点击删除按钮 | Anish | 删除成功 | 通过 |
| 修改员工 | y2-003 | 点击表格 | Anish->A | 修改成功 | 通过 |
| 查询员工 | y2-004 | 输入员工编  号或姓名，点击查询  按钮 | 220618545 | 查询成功 | 通过 |

表5.3 客户管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 增加客户 | y3-001 | 点击添加按  钮，输入‘姓  名’，‘性别’，‘出生日期’，‘电话号码’，‘会员’ | 啦啦啦，女，2022-06-22，12345678901，是 | 新增成功 | 通过 |
| 删除客户 | y3-002 | 点击删除按钮 | 啦啦啦 | 删除成功 | 通过 |
| 修改客户 | y3-003 | 点击表格 | 女->男 | 修改成功 | 通过 |
| 查询客户 | y3-004 | 输入员工编  号或姓名，点击查询  按钮 | 220622787 | 查询成功 | 通过 |

表5.4 菜品管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 增加菜品 | y4-001 | 点击添加按  钮，输入‘菜品名’，‘价格’，‘品种分类’，‘口味分类’，‘烹饪方式’ | 2313，50，家禽类，酱香味，烧 | 新增成功 | 通过 |
| 删除菜品 | y4-002 | 点击删除按钮 | 2313 | 删除成功 | 通过 |
| 修改菜品 | y4-003 | 点击表格 | 50->40 | 修改成功 | 通过 |
| 查询菜品 | y4-004 | 输入菜品编  号或菜名，点击查询  按钮 | D220623595 | 查询成功 | 通过 |

表5.5 餐台管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 增加餐台 | y5-001 | 点击添加按  钮，输入‘座位数’ | 10 | 新增成功 | 通过 |
| 删除餐台 | y5-002 | 点击删除按钮 | T220623148 | 删除成功 | 通过 |
| 修改餐台 | y5-003 | 点击表格 | 10->20 | 修改成功 | 通过 |
| 查询餐台 | y5-004 | 输入菜品编  号或菜名，点击查询  按钮 | T220623148 | 查询成功 | 通过 |

表5.6 点菜管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 开台 | y6-001 | 输入客户id和人数 | 220622545，1 | 开台成功 | 通过 |
| 生成订单 | y6-002 | 点菜后点击保存 | 无 | 操作成功 | 通过 |

表5.7 结账管理测试用例

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 功能 | 用例编号 | 操作步骤  （输入条件） | 测试数据 | 预期结果 | 实际结果/判定 |
| 删除订单 | y7-001 | 点击订单后点击删除按钮 | Z220626767 | 删除成功 | 通过 |
| 结账操作 | Y7-002 | 点击订单后输入金额 | 500 | 支付成功 | 通过 |

# 附录

## 附录A 本系统的组成

Entity

Function

UI

## 附录B 程序源代码(经过注释的关键代码)



