

《数据库系统原理》

|  |  |  |
| --- | --- | --- |
| **作业名称：** | **需求规格说明书** | |
| **姓 名：** | | **戴斌斌** | |
| **学 号：** | | **20281239** | |
| **专 业：** | | **物联网工程2001** | |

2023年6月17日

目录

[一、统一建模语言 3](#_Toc135173818)

[1.1 UML 3](#_Toc135173819)

[1.2用例（Use Case） 3](#_Toc135173820)

[1.3用况(Use Case Scenario) 3](#_Toc135173821)

[1.4 用例图(Use Case Diagram) 3](#_Toc135173822)

[1.5泳道图(Swimlane Diagram) 3](#_Toc135173823)

[1.6数据流图（Data Flow Diagram，DFD） 3](#_Toc135173824)

[二、Visio中绘制方法 4](#_Toc135173825)

[三、系统业务需求分析 5](#_Toc135173826)

[四、系统业务场景分析 6](#_Toc135173827)

[五、数据字典 6](#_Toc135173828)

[六、非功能性需求分析 9](#_Toc135173829)

# 一、统一建模语言

## 1.1 UML

统一建模语言（Unified Modeling Language，简称UML）是一种用于软件系统设计和开发的图形化建模语言。它提供了一套标准化的符号和语法，用于描述系统的结构、行为和交互。

## 1.2用例（Use Case）

用例是描述系统功能需求的一种技术手段。它表示系统与用户或其他系统之间的交互，描述了系统提供给用户或外部实体的一项功能。用例通常以用户的角度来定义，描述了用户与系统之间的目标和场景。

## 1.3用况(Use Case Scenario)

用况是对用例的进一步细化，描述了用例的具体执行过程和交互步骤。用况场景包括了用户的行为、系统的响应以及可能的结果。通过用况场景，可以更详细地了解用例的执行流程和各个参与者之间的交互。

## 1.4 用例图(Use Case Diagram)

用例图是UML中描述系统功能需求的图形化表示方式。它通过使用椭圆形表示用例，使用参与者（Actor）表示用户或外部实体，并使用箭头表示参与者与用例之间的交互关系。用例图能够帮助我们更好地理解系统的功能，并与相关利益相关者进行有效的沟通。

## 1.5泳道图(Swimlane Diagram)

道图是一种用于描述业务流程或系统流程的图形化表示方法。它使用泳道（Swimlane）来表示参与者、部门或角色，并通过排列不同泳道中的活动来展示系统的工作流程和责任分配。泳道图能够清晰地显示系统中各个参与者的角色和活动，并帮助我们理解业务流程的执行顺序和依赖关系。

## 1.6数据流图（Data Flow Diagram，DFD）

数据流图是一种用于描述系统数据流动和处理过程的图形化工具。它以圆圈（称为过程）表示数据的处理和转换，使用箭头表示数据的流动方向，以及使用矩形（称为数据存储）表示数据的存储位置。数据流图能够帮助我们分析和理解系统中数据的流动路径，以及不同组件之间的数据依赖关系。

# 二、Visio中绘制方法

在中我们在模板类的界面点击搜索UML，可以看到如下模板供我们选择：

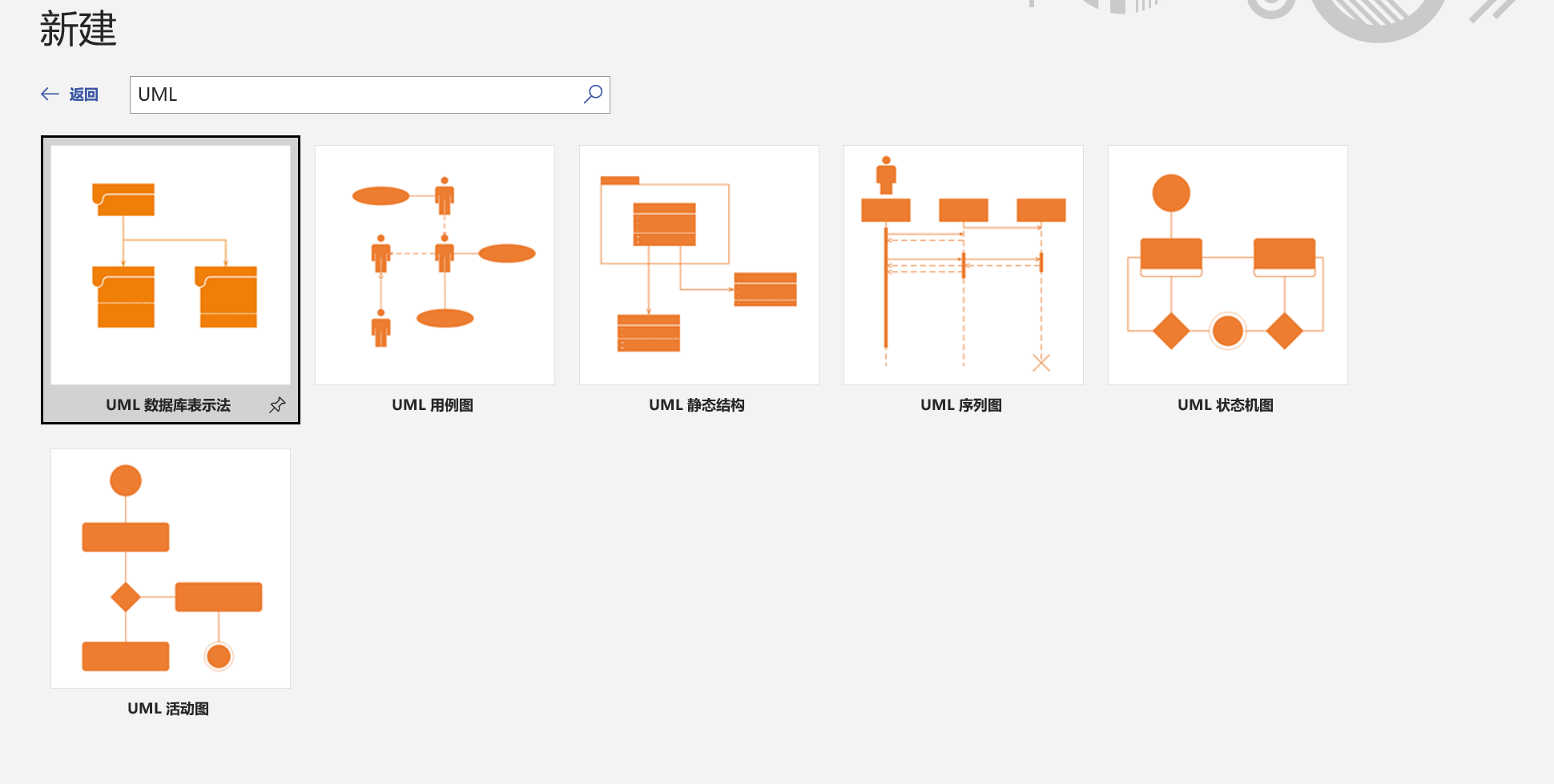


Figure Visio中UML模板

我们点击选择需要的模板进入绘制：

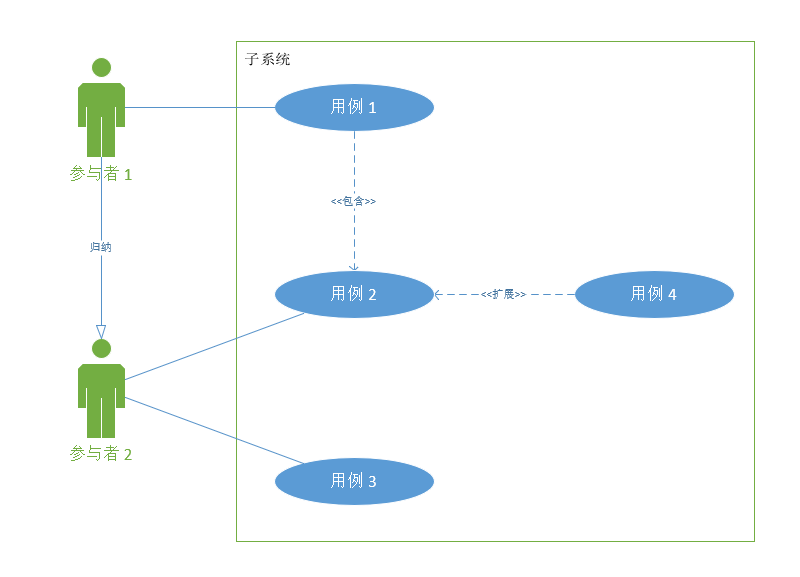


Figure UML绘制案例

在右侧我们可以选择需要的图形进行添加：

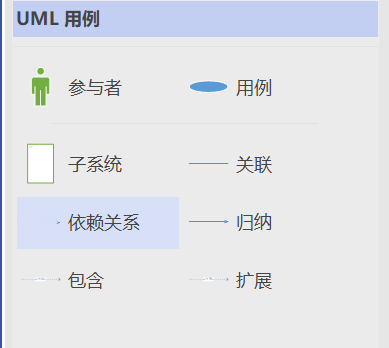


Figure 添加图形

# 三、系统业务需求分析

我们将用户划分为未登录、顾客、商家、管理员，对于上述用户有着不同的操作需求。

未登录：进行登录以及注册。

顾客：浏览商户菜品，下单

商家：增加菜品，修改菜品

会员：优惠下单，管理员：查看数据、平台维护

标识出用户(Actor或User)，进行系统功能划分，画出用例图，如下

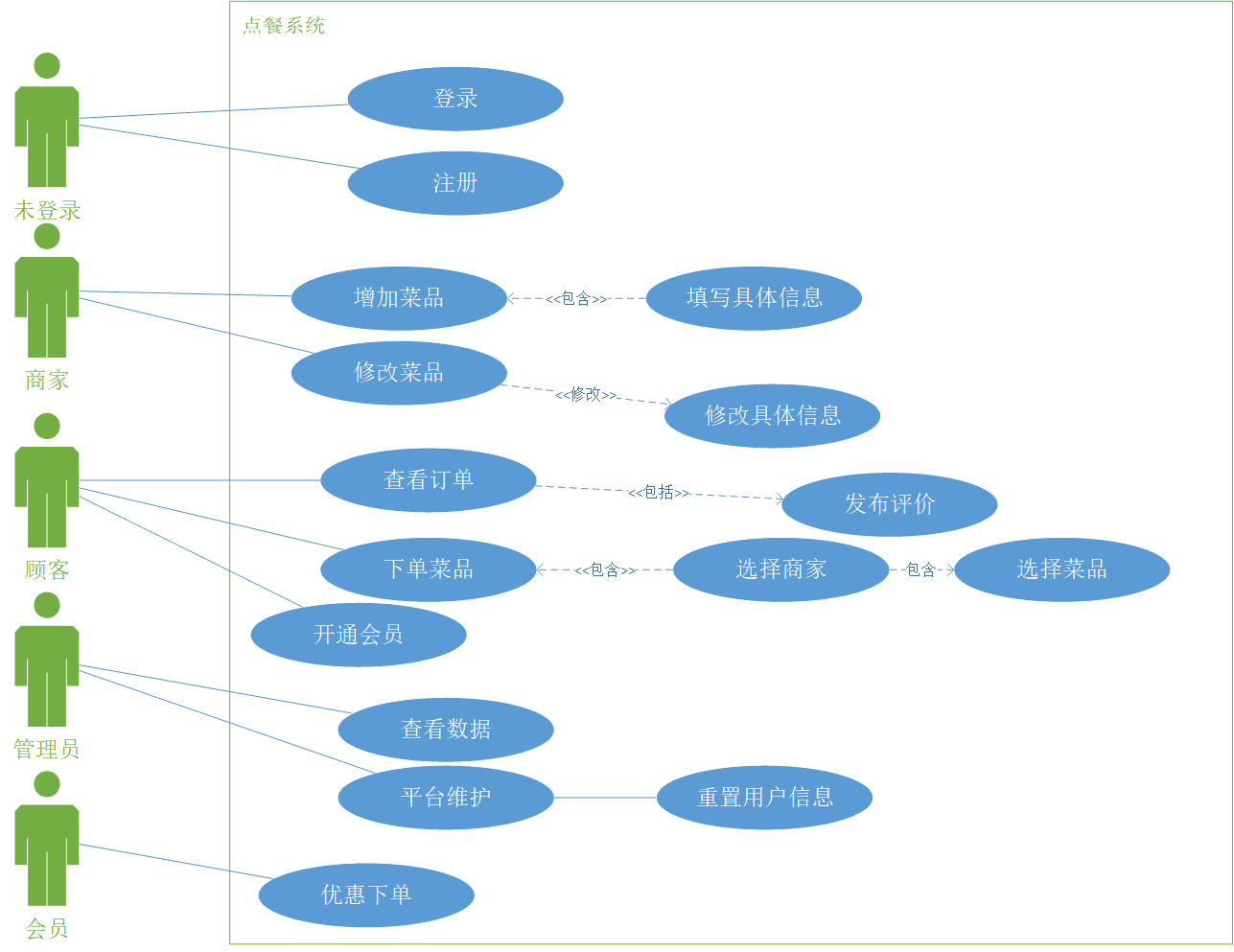


Figure 用例图

# 四、系统业务场景分析



Figure 泳道图

# 五、数据字典

数据结构（可以用类图表示）如下图所示

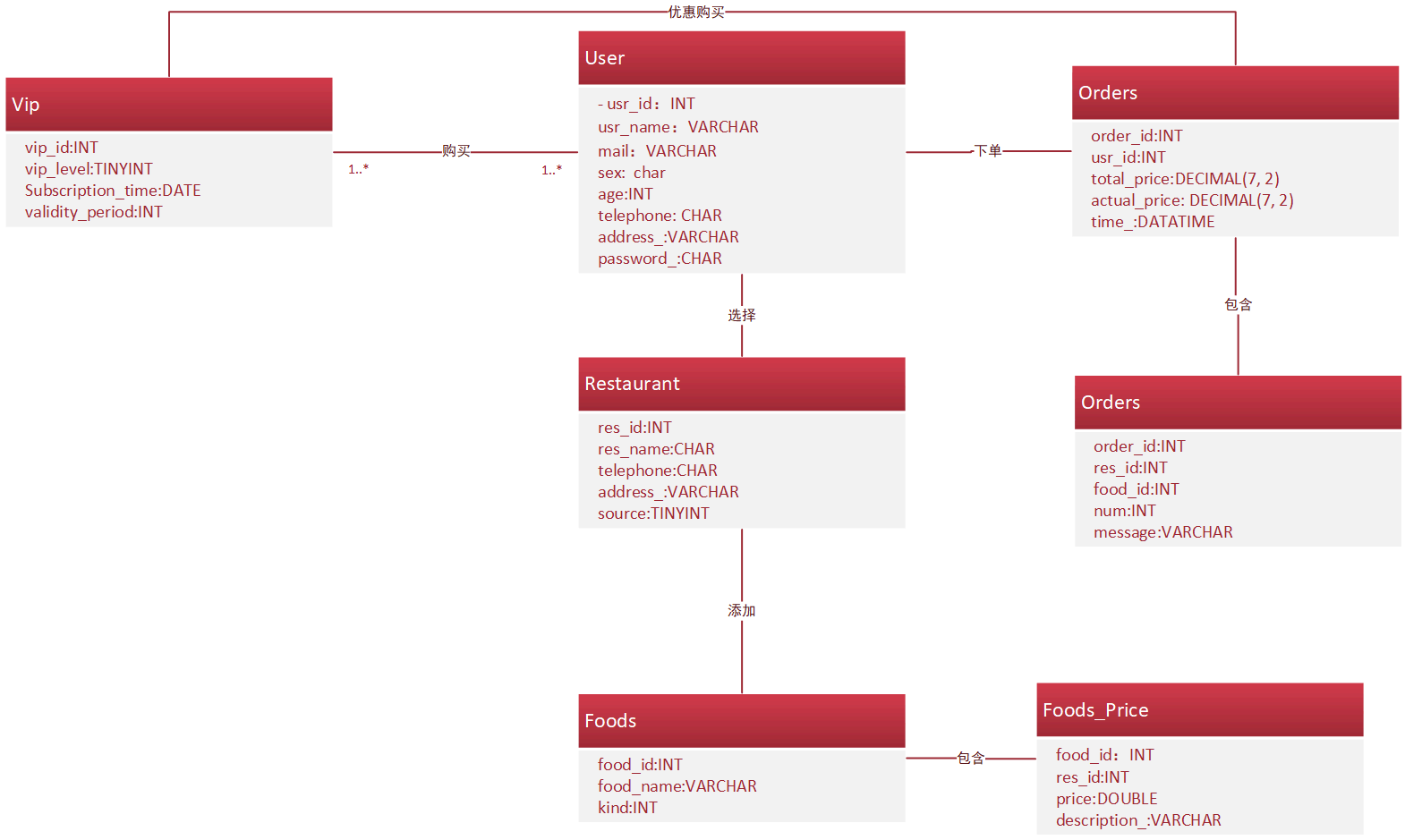


Figure 类图

数据字典如下表：

用户表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 允许为空 | 唯一 |
| usr\_id | INT |  | 否 | 是 |
| usr\_name | VARCHAR |  | 否 | 否 |
| mail | VARCHAR |  | 否 | 是 |
| sex | CHAR |  | 否 | 否 |
| age | INT |  | 否 | 否 |
| telephone | CHAR |  | 否 | 是 |
| address\_ | VARCHAR |  | 否 | 否 |
| password\_ | CHAR |  | 否 | 否 |

餐厅表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 允许为空 | 唯一 |
| res\_id | INT |  | 否 | 是 |
| res\_name | CHAR |  | 否 | 否 |
| telephone | CHAR |  | 否 | 是 |
| address\_ | VARCHAR |  | 否 | 否 |
| source | TINYINT |  | 否 | 否 |

会员表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 允许为空 | 唯一 |
| vip\_id | INT |  | 否 | 是 |
| vip\_level | TINYINT |  | 否 | 否 |
| Subscription\_time | DATE |  | 否 | 否 |
| validity\_period | INT |  | 否 | 否 |

订单表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 允许为空 | 唯一 | |
| order\_id | INT |  | 否 | 是 | |
| usr\_id | INT |  | 否 | 否 |
| total\_price | DECIMAL |  | 否 | 否 |
| actual\_price | DECIMAL |  | 否 | 否 |
| time\_ | DATETIME |  | 否 | 否 |

订单详情表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 允许为空 | 唯一 |
| order\_id | INT |  | 否 | 是 |
| res\_id | INT |  | 否 | 是 |
| food\_id | INT |  | 否 | 是 |
| num | INT |  | 否 | 否 |
| message | VARCHAR |  | 否 | 否 |

食品表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 允许为空 | 唯一 |
| food\_id | INT |  | 否 | 是 |
| food\_name | VARCHAR |  | 否 | 否 |
| kind | INT |  | 否 | 否 |

食品价格表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 默认值 | 允许为空 | 唯一 |
| food\_id | INT |  | 否 | 是 |
| res\_id | INT |  | 否 | 是 |
| price | DOUBLE |  | 否 | 否 |
| description\_ | VARCHAR |  | 否 | 否 |

# 六、非功能性需求分析

非功能性需求是系统开发中除了功能需求之外的其他要求，我们需要了解系统的性能、安全性、可靠性、可用性等的需求。

（1）系统业务处理性能需求：

1. 吞吐量：系统应能够处理大量的订单和数据流量，确保高效的业务处理。
2. 响应时间：系统应该能够在合理的时间内响应用户的请求，例如点餐，下单，订单搜索等操作，需要在短时间内完成。
3. 并发处理能力：系统需要支持同时处理多个用户的的操作请求

（2）安全性需求：

访问控制：只有合法的用户能够进入系统

数据保护：用户个人信息、支付信息等敏感数据需要进行加密存储和传输，确保数据的保密性和完整性。

（3）完整性

数据准确：数据展示正确，能够从数据库中将数据正确展示给用户。

数据一致：在用户的并发操作过程中，，可以正确的对数据进行读写操作。

可靠性和可用性：

（4）可靠性

数据不会无故丢失，当出现故障时，数据能回滚到故障之前的节点。

拥有数据操作日志，可查看数据操作。采用备份的方式，保证出现数据无法回滚时数据的正确恢复。