

《数据库系统原理》课程设计 系统实现报告

题目名称： 汇食订餐数据库系统

学号及姓名： 梁远志（15231099）

2017 年 12 月 24 日

一. 系统功能需求分析

汇食订餐系统（以下称 COLP）是一个基于 B/S 架构的产品设计。系统由作者一人独立完成，选用 Microsoft SQL Server 作为数据库，.Net Core (ASP.Net) 作为服务器后端语言；前端使用 Yahoo PureCSS 框架。网站实现了完全的前后端分离。

系统面向的使用用户有三种：

1. 管理员
2. 订餐用户
3. 餐饮商家

系统为上述三种群体设计分别的用户界面：

管理员：商家进驻；

订餐用户：查看商家列表、查看商家菜单、提交订单、查看订单、查看修改个人资料、发表查看评论；

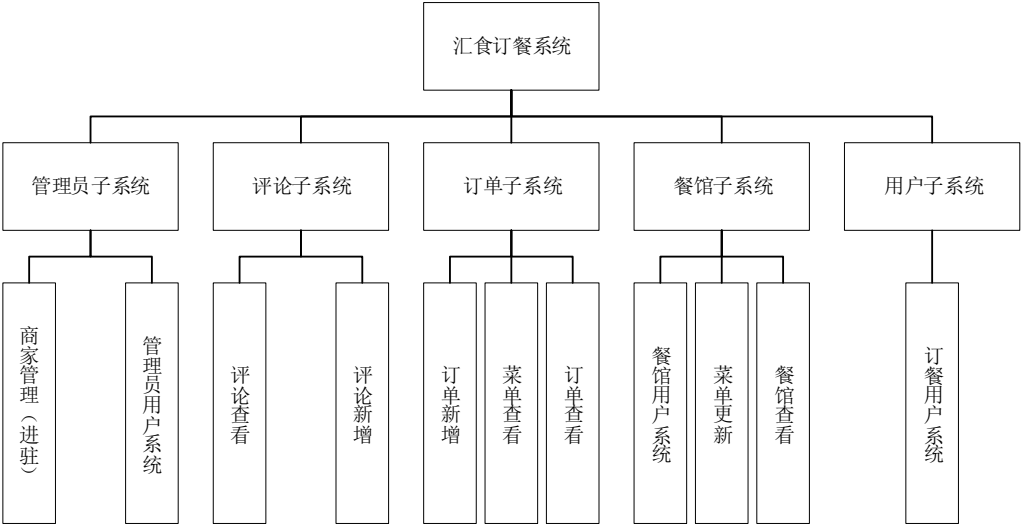
餐饮商家：查看修改菜单、查看修改店铺资料、查看订单。

二. 系统功能结构设计

结合以上需求，将系统功能设计分为五个模块：

- 1. 管理员模块：管理员的登录登出、商家进驻
- 2. 评论模块：订餐用户的发表评论、查看评论
- 3. 订单模块：订餐用户查看店家资料菜单、提交订单查看订单
- 4. 餐馆模块：餐馆用户的登录登出、注册、查看更改信息、查看更新菜单、查看订单、订餐用户查看餐馆列表
- 5. 用户模块：订餐用户的登录登出、注册、查看更改信息

系统的功能结构图如下：



注：系统实现时，虽然借鉴了 MVC 的思想，但是仅保留其控制器（Controller）的部分，上述模块分别对应实现中的控制器。

三、数据库基本表的定义

表 Comment（评价）

| 列 | 类型 | 允许 Null 值 | 说明 |
|-----------|----------|-----------|--------------|
| <u>ID</u> | bigint | | 主键，评价编号，标识规范 |
| UID | bigint | | 外键，用户编号 |
| RID | bigint | | 外键，商家编号 |
| [content] | ntext | | 评价内容 |
| datetime | datetime | | 评价时间 |

表 Menu（菜单）

| 列 | 类型 | 允许 Null 值 | 说明 |
|-------------|---------------|-----------|-----------------|
| <u>ID</u> | bigint | | 主键，菜品编号，标识规范 |
| RID | bigint | | 外键，商家编号 |
| name | nvarchar(50) | | 菜品名称 |
| description | nvarchar(100) | √ | 菜品描述 |
| price | money | | 菜品价格 |
| [order] | int | √ | 实现保留（用于排序和状态标记） |

表 Order（订单）

| 列 | 类型 | 允许 Null 值 | 说明 |
|-----------|----------|-----------|--------------|
| <u>ID</u> | bigint | | 主键，菜单编号，标识规范 |
| RID | bigint | | 外键，商家编号 |
| UID | bigint | | 外键，用户编号 |
| amount | money | | 订单总价 |
| datetime | datetime | | 订单时间 |

表 OrderContent（包含）

| 列 | 类型 | 允许 Null 值 | 说明 |
|-----------------|---------------|-----------|---------|
| <u>OID</u> | <u>bigint</u> | | 外键，订单编号 |
| <u>MID</u> | <u>bigint</u> | | 外键，商家编号 |
| <u>quantity</u> | <u>int</u> | | 数量 |

表 Restaurant（餐饮商家）

| 列 | 类型 | 允许 Null 值 | 说明 |
|--------------------|----------------------|-----------|--------------|
| <u>ID</u> | <u>bigint</u> | | 主键，商家编号，标识规范 |
| <u>name</u> | <u>nvarchar(50)</u> | | 名称 |
| <u>description</u> | <u>nvarchar(100)</u> | √ | 描述 |
| <u>username</u> | <u>varchar(50)</u> | | 用户名 |
| <u>password</u> | <u>varchar(50)</u> | | 密码 |
| <u>type</u> | <u>nvarchar(50)</u> | √ | |

表 User（订餐用户）

| 列 | 类型 | 允许 Null 值 | 说明 |
|-----------------------|---------------------|-----------|--------------|
| <u>ID</u> | <u>bigint</u> | | 主键，商家编号，标识规范 |
| <u>username</u> | <u>varchar(50)</u> | | 用户名 |
| <u>password</u> | <u>varchar(50)</u> | | 密码 |
| <u>deliverAddress</u> | <u>nvarchar(50)</u> | √ | 送餐地址 |
| <u>phone</u> | <u>varchar(50)</u> | √ | 手机号码 |

DDL 语句

```

CREATE TABLE User(
    ID bigint IDENTITY(1000,1) NOT NULL,
    username varchar(50) NOT NULL,
    password varchar(50) NOT NULL,
    deliverAddress nvarchar(50) NULL,
    phone varchar(50) NULL
);

CREATE TABLE Restaurant(
    ID bigint IDENTITY(1000,1) NOT NULL PRIMARY KEY,
    name nvarchar(50) NOT NULL,
    description nvarchar(100) NULL,
    username varchar(50) NOT NULL,
    password varchar(50) NOT NULL,
    type nvarchar(50) NULL
);

CREATE TABLE Comment(
    ID bigint IDENTITY(1000,1) NOT NULL PRIMARY KEY,
    UID bigint NOT NULL FOREIGN KEY REFERENCES User (ID),
    RID bigint NOT NULL FOREIGN KEY REFERENCES Restaurant (ID),
    [content] nvarchar(200) NOT NULL,
    [datetime] datetime NOT NULL DEFAULT (getdate())
);

CREATE TABLE Menu(
    ID bigint IDENTITY(1000,1) NOT NULL PRIMARY KEY,
    RID bigint NOT NULL FOREIGN KEY REFERENCES Restaurant (ID),
    name nvarchar(50) NOT NULL,
    description nvarchar(100) NULL,
    price money NOT NULL,
    order int NULL
);

CREATE TABLE Order(
    ID bigint IDENTITY(1000,1) NOT NULL PRIMARY KEY,
    RID bigint NOT NULL FOREIGN KEY REFERENCES Restaurant (ID),
    UID bigint NOT NULL FOREIGN KEY REFERENCES User (ID),
    [amount] money NOT NULL DEFAULT (0),

```

```
        [datetime] datetime NOT NULL DEFAULT (getdate())
    );

CREATE TABLE OrderContent(
    OID bigint NOT NULL FOREIGN KEY REFERENCES Order (ID),
    MID bigint NOT NULL FOREIGN KEY REFERENCES Menu (ID),
    quantity int NOT NULL,
    PRIMARY KEY (OID, MID)
)
```

说明：

1. DDL 语句中的方括号为标识符保留符号，用于与内置保留字冲突的情况；
2. IDENTITY 为标识规范（自增属性），括号内一次为起始数值和自增长度；
3. nvarchar 为宽字符变长字符串类型，最多容纳的是字符（默认为 UTF8）个数；
4. DEFAULT 指定了该列在没有显式插入赋值的缺省值，getdate()为当前时间；

四、触发器的定义与实现

系统针对系统的两个主要事务过程设计了针对数据校验的触发器：

CheckMenuPrice 触发器（餐品价格检查）：

```
CREATE TRIGGER CheckMenuPrice
ON [Menu]
FOR INSERT
AS
DECLARE @price MONEY;
SELECT @price=[price] FROM INSERTED;
IF (@price > 100 OR @price <= 0)
BEGIN
    RAISERROR('Invalid menu item price!', 16, 11);
    ROLLBACK TRANSACTION;
END
```

触发器功能：在商家更改订单提交时（一个事务），如果有插入的单项菜品的价格不在约束范围内，则报出错误，同时触发事务的回滚。

CheckOrderItem 触发器（订单数量检查&总价更新）：

```
CREATE TRIGGER CheckOrderItem
ON [OrderContent]
FOR INSERT
AS
DECLARE @quantity INT;
DECLARE @MID BIGINT;
DECLARE @OID BIGINT;
SELECT @quantity=[quantity], @MID=[MID], @OID=[OID] FROM INSERTED;
IF (@quantity > 100 OR @quantity <= 0)
BEGIN
    RAISERROR('Invalid item quantity!', 16, 11);
    ROLLBACK TRANSACTION;
END

DECLARE @amount MONEY;
SELECT @amount=([price] * @quantity) FROM [Menu] WHERE [ID]=@MID;
UPDATE [Order] SET [amount]=[amount] + @amount WHERE [OID]=@OID;
```

触发器功能：在用户提交一个订餐时（一个事务），如果单项餐品的数量不在约束范围内，则报出错误；同时会累加计算订单总价，更新 Order 表中的总价值。

五、存储过程的定义与实现

系统定义了对数据进行更新或插入的若干存储过程用于简化数据插入的语句：

UserUpdate 存储过程（用户更新个人资料）

```
CREATE PROCEDURE UserUpdate
    @username varchar(50),
    @password varchar(50),
    @deliverAddress nvarchar(50),
    @phone varchar(50)
AS
UPDATE [User]
SET [password]=@password,
    [phone]=@phone,
    [deliverAddress]=@deliverAddress
WHERE [username]=@username
```

UserRegister 存储过程（用户注册）

```
CREATE PROCEDURE UserRegister
    @username varchar(50),
    @password varchar(50),
    @deliverAddress nvarchar(50) NULL = NULL,
    @phone varchar(50) NULL = NULL
AS
INSERT INTO [User]
([username], [password], [deliverAddress], [phone])
VALUES
(@username, @password, @deliverAddress, @phone)
```

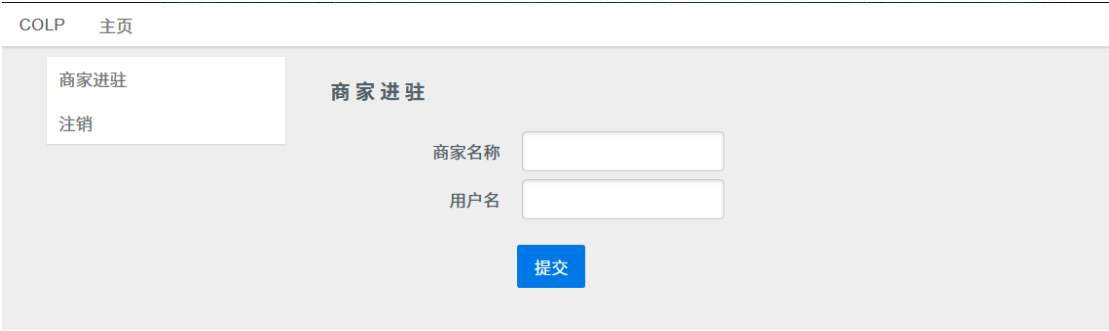
RestaurantRegister 存储过程（商家注册）

```
CREATE PROCEDURE RestaurantRegister
    @name nvarchar(50),
    @username varchar(50),
    @password varchar(50)
AS
INSERT INTO [Restaurant]
([name], [username], [password])
VALUES (@name, @username, @password)
```

此外在设计系统时也遇到了需要使用事务技术来进行的较大规模的数据操作：修改菜单、提交订单。又因为其参数不定长，难以使用特定存储过程来进行事务执行。所以只有依托于 ADO.net 提供的事务执行接口来完成出错的回滚以及触发器的触发。具体情况请参看源代码。

六、系统实现结果

系统已经部署于作者的服务器上，如需直接访问，可访问 <http://konoha.lyzde.com/>，（或 <http://bus.lyzde.com/>）管理员账户密码都为 admin。以下是一些系统截图：



选择您要就餐的餐馆类型

全部

云南菜

川菜

日韩料理

江浙菜

火锅

粤菜

西北菜

西式快餐

云海肴

超级好吃，黑三剁配米饭完美

查看菜单

南京大牌档

去过很多次啦，一直都很喜欢

查看菜单

日昌餐馆

煲仔饭很好吃，儿子赞不绝口

查看菜单

汉堡王

套餐价位不错，人流量挺多的

查看菜单

沸腾鱼乡

环境还可以，值得推荐榴莲酥

查看菜单

海底捞火锅

川味火锅为主、融汇各地火锅特色为一体的大型跨省直营餐饮品牌火锅店

查看菜单

西北莜面村

菜品味道很好，服务也很周到

查看菜单

青山日本料理

很好吃啊

查看菜单

COLP 主页 餐馆 我的帐号

资料

订单

注销

个人资料

用户名

tonny

密码

手机号码

12121212

送餐地址

花园街道学院路37号北京

提交

订 单

餐馆 云海肴

时间 2017-11-16T21:37:24.437

金额 ¥338.6

内容

- 黑三剁 ¥16.8 * 2
- 鸡丝凉米线 ¥36 * 2
- 傣味香茅草烤罗非鱼 ¥62 * 1
- 原味汽锅鸡 ¥23 * 1
- 包烧野生牛肝菌 ¥55 * 2
- 香煎石屏豆腐 ¥13 * 2
- 黑糖豆花 ¥12 * 1

- 资料
- 菜单
- 订单
- 注销

资 料

商家名称

云海肴

用户名

003665

密码

003

描述

超级好吃，黑三剁配米线

类型

云南菜

提交

菜 单

新建菜品

名称

描述

价格

新增

| | | | |
|---------------------|--------|----|----|
| 黑三剁 黑三剁 | ¥ 16.8 | 删除 | 修改 |
| 鸡丝凉米线 鸡丝凉米线 | ¥ 36 | 删除 | 修改 |
| 傣味香茅草烤罗非鱼 傣味香茅草烤罗非鱼 | ¥ 62 | 删除 | 修改 |
| 原味汽锅鸡 原味汽锅鸡 | ¥ 23 | 删除 | 修改 |
| 包烧野生牛肝菌 包烧野生牛肝菌 | ¥ 55 | 删除 | 修改 |
| 香煎石屏豆腐 香煎石屏豆腐 | ¥ 13 | 删除 | 修改 |
| 黑糖豆花 黑糖豆花 | ¥ 12 | 删除 | 修改 |

保存更改

订 单

送餐地址 花园街道学院路37号北京航空航天大学
时间 2017-11-16T21:37:24.437
金额 ¥338.6

云海肴

超级好吃，黑三剁配米饭完美

商家菜单

| # | 菜品名称 | 价格 | 数量 |
|---|---------------------------|-------|-------|
| 1 | 黑三剁 | ¥16.8 | - 2 + |
| 2 | 鸡丝凉米线 | ¥36 | - 2 + |
| 3 | 傣味香茅草烤罗非鱼 | ¥62 | - 1 + |
| 4 | 原味汽锅鸡 | ¥23 | - 1 + |
| 5 | 包烧野生牛肝菌 | ¥55 | - 1 + |
| 6 | 香煎石屏豆腐 | ¥13 | - 0 + |
| 7 | 黑糖豆花 | ¥12 | - 0 + |

用户评价

tonny在2017-11-16T21:41:05.09评论

这家的菠萝饭挺好吃的，强烈安利！

新建评论

提交

结算

当前订单总价

¥ 245.6

您的送餐地址

花园街道学院路
37号北京航空航
天大学

订单预计在半小时后送达

提交订单

七、总结

这个项目虽然在实现上费的功夫远远大于其设计上，尽管作者拥有比较丰富的 Web 开发经验。不过不得不说的是，这是第一次按照一个数据库系统设计应有的流程一步一步走下来的，这个过程中我收获还是很多的。

从需求分析到数据流图、到数据元素表、到 E-R 图、到关系数据模型这一系列的“理论”上的工作不仅能够巩固在理论课上学习到的知识，同时也如神来之笔一般对整个数据库系统的基础设计起到了具有决定性的推进作用。事实上我发现任何在实现时发现的设计失误都能够追溯到这几个设计步骤上，所有的流程看上去都显得那么的有理有据。

在实现这个数据库系统的时候，也充分的提高了我的 SQL 编程能力。虽然在有些地方有点牵强，但是在这些方面（DBMS 提供的用户定义约束（外键、唯一性）、存储过程设计、触发器设计、事务过程调用）在体验到由 DBMS 带来的数据操作的便利性的同时也能从功能的角度理解这些机构设定的目的。

由于时间仓促，系统在很多地方的设计并不那么让人满意，但是对于我对数据库设计以及 DBMS 的使用的提高是切实可见的。