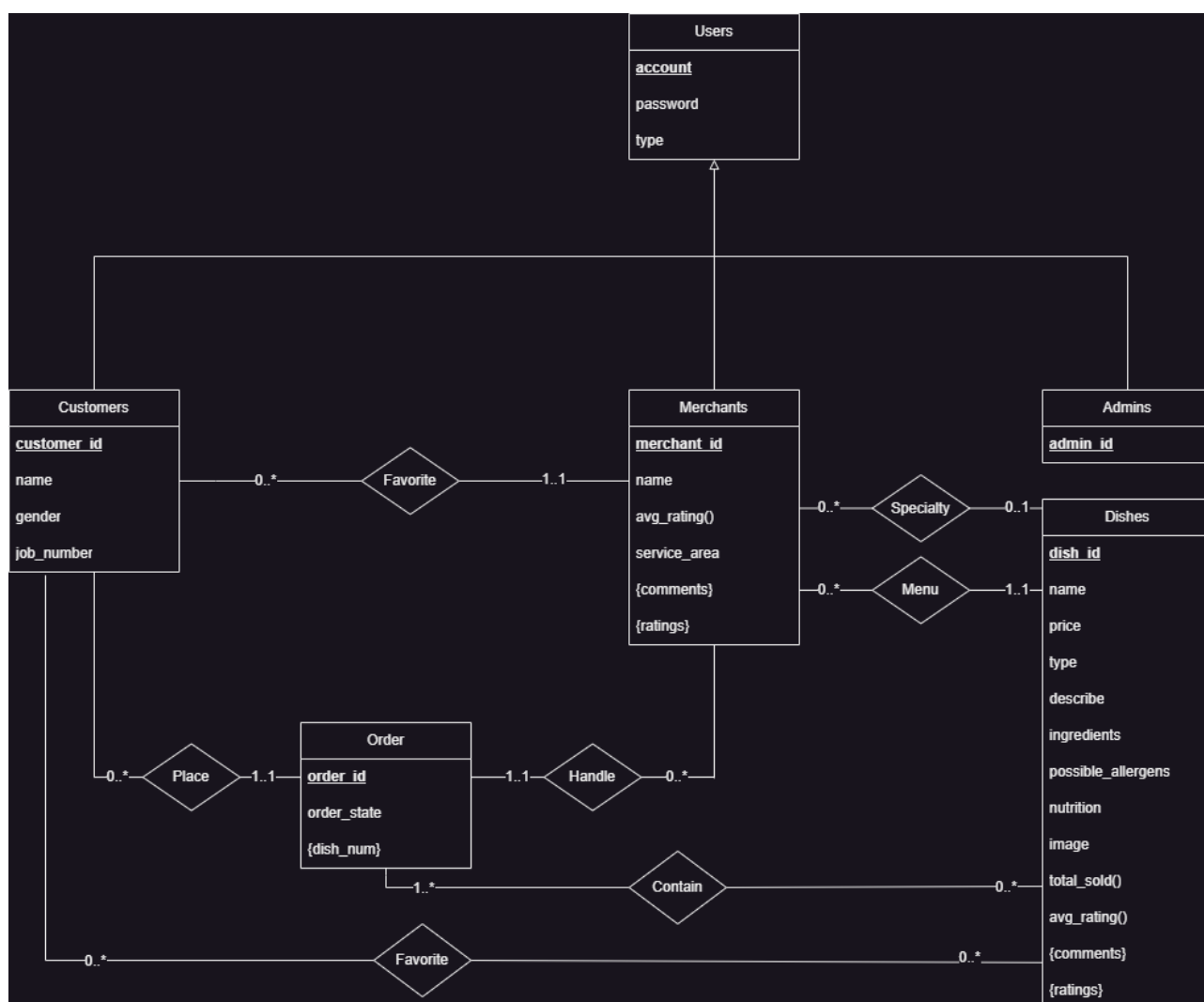


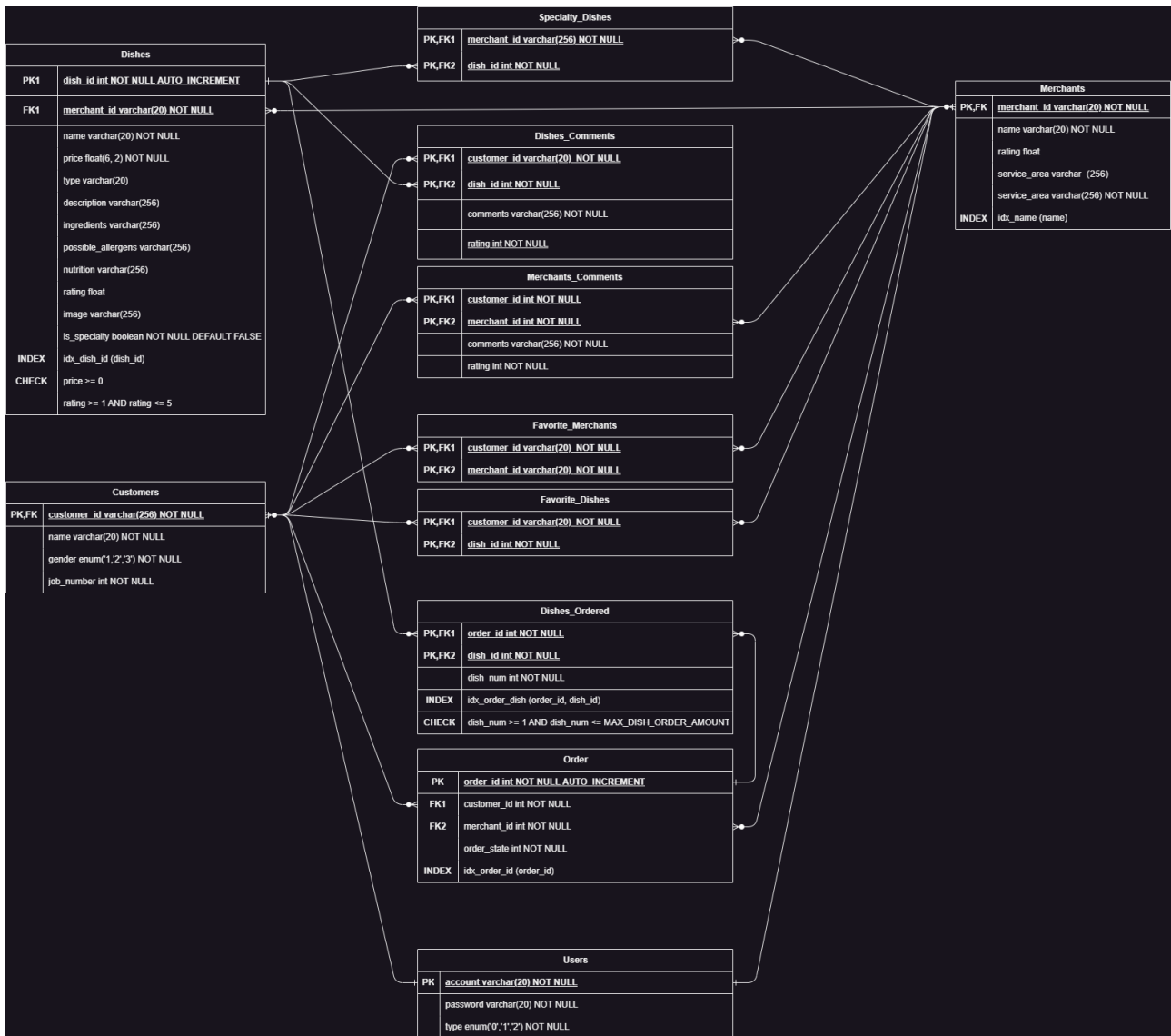
一、E-R图和表结构

我们小组的E-R图和表结构较之于中期提交的版本又有一些变化，在修改后如下：

E-R图：



表结构：



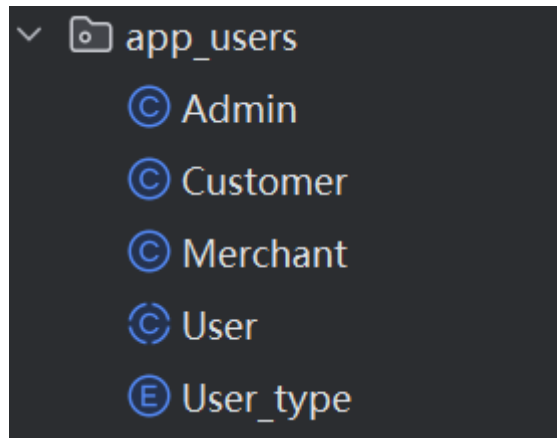
二、代码结构

代码分包如下：



其中，**app_operation**中的类处理食堂点餐程序与用户的交互动作；**app_user**是该系统用户层功能的实现（管理员、顾客、商户）集合；**customException**是自定义异常类的集合，用于处理程序中的异常情况；**database_operation**是数据库操作相关类的集合，包括了数据库链接、增删改查关系等诸多功能；**utility**是工具类集合，包括常量类等。通过这样的代码结构设计，我们实现了对数据库操作和对用户功能实现的大致隔离，通过清晰的职责分离，我们使自己的代码更易于理解和维护，还提高了代码重用性和小组成员的协作效率。

对于app_users，我们设计了一个枚举类User_Type,一个抽象类User及其三个子类Admin、Customer、Merchant，分别代表了管理员、顾客和商家。



三、项目亮点

- 我们创建了许多自定义异常类，对于各种边界条件或其他异常都有较好的处理，在发生SQL错误时，不会直接把SQL错误返回给用户。
- 我们对SQL注入进行了防范，不仅采用了 **Prepared Statement**，也对其他输入进行了严格的格式判断与处理，在app_users中，我们没有使用任何scanner的方法，取而代之的是在User_interface中的包装的输入方法，这些方法在格式、长度等方法对输入字符串进行检查，实现了用户友好的程序交互。
- 在建表时，我们采用了索引等方法加快数据库的查询，提高大数据规模下的操作性；使用了级联删除的方法，在Dishes的实例中，当Merchant账号被删除，其创建的Dishes也同步被删除；此外，我们也使用了SQL的完整性约束（CHECK）来保证数据库中的数据正确性。以下是Dishes的建表示例：

```
stmt.executeUpdate("create table if not exists
Dishes(" +
    "dish_id int NOT NULL AUTO_INCREMENT," +
    "merchant_id varchar(20) NOT NULL," +
    "name varchar(20) NOT NULL," +
    "price float(6,2) NOT NULL," +
    "type varchar(20)," +
    "description varchar(256)," +
    "ingredients varchar(256)," +
    "possible_allergens varchar(256)," +
    "nutrition varchar(256)," +
    "rating float," +
    "image varchar(256)," +
```

```

        "is_specialty boolean NOT NULL DEFAULT
FALSE," +
        "foreign key(merchant_id) references
Merchants(merchant_id) on delete cascade," +
        "primary key(dish_id)," +
        "index idx_dish_id (dish_id)," +
        "CHECK (price >= 0)," +
        "CHECK (rating >= 1 AND rating <= 5)" +
        ");");

```

在这里，我们在dish_id这一经常被搜索的属性上建立了索引，以加快查找速度。

- 我们尝试使用了视图这一SQL特性，在下面的视图中，展示了一个菜品的简要信息：

```

stmt.executeUpdate("create or replace view
simpleDishview AS " +
        "SELECT dish_id,merchant_id,name,price,type
" +
        "FROM Dishes;");

```

但是，正如我们在后面的项目缺陷的分析中提及的一样，由于本项目建立在同一个数据库user id下，实际上视图对于数据获取的控制作用几乎没有，反而会增加不必要的开销，所以，我们对于视图的使用只是一个尝试。

- 我们在系统中加入了分页询问功能，已顾客查看菜单为例(此处将分页常数即一页显示的最大量MAX_DISPLAY_NUM修改为2便于展示)：

请选择需要进行的操作(输入8查看所有指令)：

1

菜单如下：

1	菜品名：梅干菜蛋炒饭	价格（元）：16.0	菜品类型：主食
---	------------	------------	---------

2	菜品名：里脊肉蛋炒饭	价格（元）：20.0	菜品类型：1
---	------------	------------	--------

列表过长，是否继续显示（Y/N）？

y

3	菜品名：配饭一绝西瓜汁	价格（元）：4.0	菜品类型：饮料
---	-------------	-----------	---------

菜单显示完毕

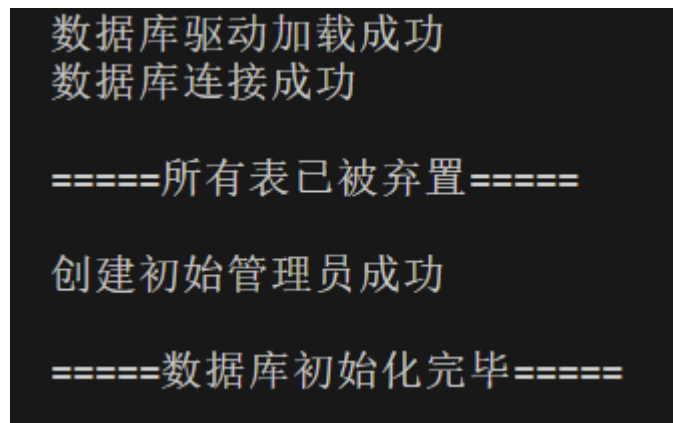
- 在SQL的数据更新、插入过程中，我们多次利用了事务的特性，以保证操作的原子性，详见第六小节的说明
- 我们允许用户通过不完整的名称模糊搜索对应的物品，再由用户进行进一步的选择，而商家、菜品的ID对于用户是隐藏的，这更符合实际情况，而且同样实现了用户友好的程序交互。

四、项目缺陷和我们的改进思路

- 实际上本项目仍是建立在同一个数据库user id下的，还暂时无法做到真正的多用户共同访问数据库，我们的想法是可以考虑在管理员创建新用户后向数据库添加新用户的账号和密码，为不同的角色赋予不同的权限，并在登出后切断当前的connection，严格地限制用户可以进行的操作，但由于时间仓促，这一改进想法未能付诸实践。
- 在本项目中仍有资源未合理关闭的代码内部质量问题，我们的改进思路是应该避免返回结果集，在finally块中关闭资源，避免泄露。
- 没有实现很好的UI图形界面

五、功能实现

第一次运行此程序时（或者启用了dropAllTables方法），会初始化各个表、视图等等。并创建初始管理员，账号为admin，密码1234（此管理员账户不可被删除）。



```
数据库驱动加载成功
数据库连接成功

=====所有表已被弃置=====

创建初始管理员成功

=====数据库初始化完毕=====
```

此后运行时，

1.三种用户的登录、登出

此处以管理员登录为例，刚刚登录，会显示一次所有指令：

```
=====请输入账号密码=====
```

```
账号:   admin
```

```
密码:   1234
```

```
=====管理员登录成功=====
```

```
-h      显示本列表
```

```
-c      创建账户
```

```
-i      查询账户信息
```

```
-s      设置账户信息
```

```
-d      删除账户信息
```

```
-di     查询所有菜品信息
```

```
-ds     查询某菜品总销量
```

```
-cf     查询所有商家的收藏情况
```

```
-q      退出账号
```

```
-exit   退出程序
```

```
以上所有指令大小写不分，且对于名称的搜索支持关键词查询  
请输入指令（输入 '-h' 查看所有指令）：  
█
```

登出（输入-q后的结果，清屏，等待其他用户的登录）：

```
=====请输入账号密码=====  
账号:  █
```

2.-h 显示指令列表

对三种用户，都可以输入-h以显示支持的指令列表

(1) 管理员

请输入指令 (输入 '-h' 查看所有指令):

```
-h
-h      显示本列表
-c      创建账户
-i      查询账户信息
-s      设置账户信息
-d      删除账户信息
-di     查询所有菜品信息
-ds     查询某菜品总销量
-cf     查询所有商家的收藏情况
-q      退出账号
-exit   退出程序
```

以上所有指令大小写不分，且对于名称的搜索支持关键词查询

(2) 商家

请输入指令 (输入 '-h' 查看所有指令):

```
-h
-h      显示本列表
-i      查询本账户信息
-s      修改本账户信息
-ad     添加菜品
-as     添加招牌菜
-ud     修改菜品信息
-dd     删除菜品信息
-m      显示菜单
-o      展示所有与自己相关的订单
-fo     完成订单
-ds     删除招牌菜
-q      退出账号
-exit   退出程序
```

以上所有指令大小写不分，且对于名称的搜索支持关键词查询

(3) 顾客

请输入指令 (输入 '-h' 查看所有指令):

```
-h
-h      显示本列表
-i      查询本账户信息
-s      设置本账户信息
-fd     收藏喜欢的菜品
-fm     收藏喜欢的商家
-sm     搜索特定商家信息
-sd     搜索特定菜品信息
-o      预订餐厅和点餐
-so     展示所有与自己相关的订单
-df     取消收藏菜品
-co     评价订单
-q      退出账号
-exit   退出程序
```

以上所有指令大小写不分，且对于名称的搜索支持关键词查询

下面将以截图形式详细地展示这三种用户的各项功能：

3.管理员

(1) -c 创建用户

请输入指令 (输入 '-h' 查看所有指令):

```
-c
请输入新账号密码及权限组(0: 管理员 1: 食堂商户 2: 顾客):
新账号: customer1
新密码: password
权限组: 2
账号创建失败，可能是因为账号已存在，请重新输入
请输入新账号密码及权限组(0: 管理员 1: 食堂商户 2: 顾客):
新账号: customer2
新密码: password
权限组: -1
非法输入！请输入以下合法输入中的一项: [0, 1, 2]
3
非法输入！请输入以下合法输入中的一项: [0, 1, 2]
2
账号成功创建
```


(2) -i 查询账号信息

1.不存在相关结果

```
请输入指令 (输入 '-h' 查看所有指令):  
-i  
请输入所需查询的账号:    customer3  
未查询到账户讯息, 请检查账户ID是否正确
```

2.账号存在但信息未设置

```
请输入指令 (输入 '-h' 查看所有指令):  
-i  
请输入所需查询的账号:    merchant1  
该账号为食堂商户  
该账户未创建信息
```

3.账号存在且信息已设置

```
请输入指令 (输入 '-h' 查看所有指令):  
-i  
请输入所需查询的账号:    merchant1  
该账号为食堂商户  
账号      : merchant1  
店铺名称   : 黄焖鸡米饭  
店铺评分   : 0.0  
店铺位置   : 政肃路
```

(3) -s 设置账号信息

```
请输入指令 (输入 '-h' 查看所有指令):  
-s  
请输入所需设置信息的账号:    merchant1  
请输入信息:  
店铺名称:    黄焖鸡米饭  
店铺地址:    政肃路  
信息设置成功
```

(4) -d 删除账号

```
请输入指令 (输入 '-h' 查看所有指令):
-d
请输入需要删除的用户id(输入-q退出):
merchant4
账号merchant4成功删除
```

(5) -di 查看所有菜品总信息

```
请输入指令 (输入 '-h' 查看所有指令):
-di
菜品ID 商家ID 名称 价格 分类 描述 原料 可能过敏原 营养成分 评分 图片 是否为招牌
1 merchant3 梅干菜蛋炒饭 16.0 主食 梅干菜拌在蛋炒饭里 梅干菜, 蛋, 饭 111 111 111 2.5 no image false 2.5 no image true
2 merchant3 里脊肉蛋炒饭 20.0 主食 多汁里脊肉搭配鸡蛋 111 111 111 2.5 no image false
3 merchant3 牛肉蛋炒饭 23.5 主食 2 22 2 2 2.5 no image true
5 merchant3 配饭一绝西瓜汁 4.0 饮品 5 5 5 5 5.0 no image false
6 merchant1 扬州炒饭 15.0 主食 2 2 2 2 2.5 no image false
```

(6) -ds 查看某一菜品销量

```
请输入指令 (输入 '-h' 查看所有指令):
-ds
菜品ID 商家ID 名称 价格 分类 描述 原料 可能过敏原 营养成分 评分 图片 是否为招牌
1 merchant3 梅干菜蛋炒饭 16.0 主食 梅干菜拌在蛋炒饭里 梅干菜, 蛋, 饭 111 111 111 2.5 no image false 2.5 no image true
2 merchant3 里脊肉蛋炒饭 20.0 主食 多汁里脊肉搭配鸡蛋 111 111 111 2.5 no image false
3 merchant3 牛肉蛋炒饭 23.5 主食 2 22 2 2 2.5 no image true
5 merchant3 配饭一绝西瓜汁 4.0 饮品 5 5 5 5 5.0 no image false
6 merchant1 扬州炒饭 15.0 主食 2 2 2 2 2.5 no image false
请输入想要查询的菜品ID(输入-q退出):
2
此菜品总销量为3
```

(7) -cf 查看商家被收藏数

在此列表中仅显示已设置过店铺信息的商家

```
请输入指令 (输入 '-h' 查看所有指令):
-cf
商家ID 收藏数
merchant1 1
merchant3 1
```

4. 商户

(1) 登录时

若商家并未设置过店铺信息，则会在登录时强制提问：

```
=====请输入账号密码=====
账号:  merchant3
密码:  password
=====食堂商户登录成功=====
账户信息未设置，请先设置账户信息
请输入信息：
店铺名称:      阿姨蛋炒饭
店铺地址:      国定路
信息设置成功，之后您可以再修改这些信息
```

(2) -i 查看本账号信息

```
请输入指令 (输入 '-h' 查看所有指令):
-i
账号      : merchant3
店铺名称   : 阿姨蛋炒饭
店铺评分   : 0.0
店铺位置   : 国定路
```

(3) -s 修改本账号信息

```
请输入指令 (输入 '-h' 查看所有指令):
-s
请输入信息：
店铺名称:      阿姨蛋炒饭
店铺地址:      国定路602号
信息修改成功
请输入指令 (输入 '-h' 查看所有指令):
-i
账号      : merchant3
店铺名称   : 阿姨蛋炒饭
店铺评分   : 0.0
店铺位置   : 国定路602号
```

(4) -ad 添加新菜品

添加新菜品时菜品ID会自动分配且递增

```
请输入指令 (输入 '-h' 查看所有指令):
-ad
请输入新菜品信息:
菜品名称:      梅干菜蛋炒饭
菜品价格:      16
菜品分类:      主食
菜品描述:      梅干菜拌在饭里
菜品原料:      梅干菜, 蛋, 饭
可能的过敏源:  梅干菜, 蛋
营养成分:      膳食纤维, 蛋白质, 淀粉
新菜品添加成功! 新菜品ID为1
```

(5) -as 添加招牌菜

1.添加新招牌菜

```
请输入指令 (输入 '-h' 查看所有指令):
-as
序号      菜品ID  菜品名称      价格
1          1      梅干菜蛋炒饭  16.00
2          2      里脊肉蛋炒饭  20.00
3          3      牛肉蛋炒饭    23.50
4          5      配饭一绝西瓜汁 4.00
请选择你想要添加作为本店招牌菜的菜品序号:
1
梅干菜蛋炒饭已成功添加为本店招牌菜!
您想再添加其他特色菜吗? (y/n)
y
请选择你想要添加作为本店招牌菜的菜品序号:
3
牛肉蛋炒饭已成功添加为本店招牌菜!
您想再添加其他特色菜吗? (y/n)
n
```

2.添加已添加过的招牌菜

请输入指令 (输入 '-h' 查看所有指令):

-as

序号	菜品ID	菜品名称	价格
1	1	梅干菜蛋炒饭	16.00
2	2	里脊肉蛋炒饭	20.00
3	3	牛肉蛋炒饭	23.50
4	5	配饭一绝西瓜汁	4.00

请选择你想要添加作为本店招牌菜的菜品序号:

1

该菜品已经是招牌菜了

您想再添加其他特色菜吗? (y/n)

n

(6) -ud 修改菜品信息

请输入指令 (输入 '-h' 查看所有指令):

-ud

序号	菜品ID	菜品名称	价格
1	1	梅干菜蛋炒饭	16.00
2	2	里脊肉蛋炒饭	20.00
3	3	牛肉蛋炒饭	23.50
4	5	配饭一绝西瓜汁	4.00

请选择你想要修改信息的菜品序号:

1

该菜品的详细信息如下:

1.菜品名: 梅干菜蛋炒饭

2.菜品类型: 主食

3.菜品描述: 梅干菜拌在饭里

4.食材: 梅干菜, 蛋, 饭

5.可能的过敏原: 梅干菜, 蛋

6.营养成分: 膳食纤维, 蛋白质, 淀粉

7.图片: no image

8.评分: 2.5

请选择你想要修改信息的菜品条目序号(评分无法修改):

3

请输入修改后的信息:

梅干菜拌在蛋炒饭里

梅干菜蛋炒饭的菜品描述已经被成功更新!

您想再修改其他菜品吗? (y/n)

n

(7) -dd 删除菜品

```
请输入指令 (输入 '-h' 查看所有指令):
-dd
序号    菜品ID    菜品名称        价格
1        1        梅干菜蛋炒饭    16.00
2        2        里脊肉蛋炒饭    20.00
3        3        牛肉蛋炒饭      23.50
4        4        配饭一绝的冬瓜茶      3.00
请输入想删除的菜品的ID(输入-q退出):
4
菜品删除成功!
```

(8) -m 展示菜单

```
请输入指令 (输入 '-h' 查看所有指令):
-m
序号    菜品ID    菜品名称        价格
1        1        梅干菜蛋炒饭    16.00
2        2        里脊肉蛋炒饭    20.00
3        3        牛肉蛋炒饭      23.50
4        5        配饭一绝西瓜汁  4.00
```

(9) -o 查看相关订单

1.不存在相关订单

```
请输入指令 (输入 '-h' 查看所有指令):
-o
当前没有相关订单
```

2.已有相关订单

```
请输入指令 (输入 '-h' 查看所有指令):
-o
1    订单号: 4    订单状态: 待出餐    商家名称: 阿姨蛋炒饭    菜品: 配饭一绝西瓜汁 1份
2    订单号: 2    订单状态: 已出餐    商家名称: 阿姨蛋炒饭    菜品: 里脊肉蛋炒饭 2份
3    订单号: 1    订单状态: 待出餐    商家名称: 阿姨蛋炒饭    菜品: 梅干菜蛋炒饭 1份 里脊肉蛋炒饭 1份
```

(10) -fo 出餐

在商家出餐后可以选择并将订单状态更改为已出餐并在顾客登录时通知顾客：

请输入指令 (输入 '-h' 查看所有指令):

-fo

```
1      订单号: 4      订单状态: 待出餐      商家名称: 阿姨蛋炒饭      菜品: 配饭一绝西瓜汁 1份
2      订单号: 1      订单状态: 待出餐      商家名称: 阿姨蛋炒饭      菜品: 梅干菜蛋炒饭 1份 里脊肉蛋炒饭 1份
输入未完成的订单的序号, 即可将其标记为已出餐
2
该订单已出餐, 待用户上线时会自动通知
```

(11) -ds 删除招牌菜

请输入指令 (输入 '-h' 查看所有指令):

-ds

序号	菜品ID	菜品名称	价格
1	1	梅干菜蛋炒饭	16.00
2	2	里脊肉蛋炒饭	20.00
3	3	牛肉蛋炒饭	23.50
4	5	配饭一绝西瓜汁	4.00

请选择你想要删除的本店招牌菜的菜品序号:

1

梅干菜蛋炒饭已从本店招牌菜中移除!

您想再删除其他招牌菜吗? (y/n)

y

请选择你想要删除的本店招牌菜的菜品序号:

2

该菜品不是招牌菜

您想再删除其他招牌菜吗? (y/n)

n

5.顾客

(1) 登录时

1.类似商家, 若登录时未设置账号信息则强制提问, 不加以展示

2.当有订单已出餐, 登录时会对顾客进行通知


```
=====请输入账号密码=====  
账号:    customer1  
密码:    password  
=====顾客登录成功=====  
您在阿姨蛋炒饭处的订单（订单号2）已出餐！  
您在阿姨蛋炒饭处的订单（订单号4）已出餐！
```

顾客输入不同指令（大小写均可），可以执行如下操作：

（2）-i 查看个人信息

```
请输入指令（输入 '-h' 查看所有指令）：  
-i  
账号      : customer1  
用户昵称      : Hcy  
性别      : 1  
学号      : 22302010011
```

（3）-s 修改本账户信息

```
请输入指令（输入 '-h' 查看所有指令）：  
-s  
请输入信息：  
用户昵称:      HCY  
性别（男:1 女:2 其他:3）:      3  
学号:    22302010011  
信息修改成功
```

（4）-fd 收藏喜欢的菜品

a.第一次收藏菜品


```
请输入指令 (输入 '-h' 查看所有指令):
-fd
请输入想要收藏的菜品名:
炒饭
符合条件的菜品信息如下:
菜品1  商户名称: 阿姨蛋炒饭      菜品名: 梅干菜蛋炒饭      价格: 16.0
菜品2  商户名称: 阿姨蛋炒饭      菜品名: 里脊肉蛋炒饭      价格: 20.0
菜品3  商户名称: 阿姨蛋炒饭      菜品名: 牛肉蛋炒饭        价格: 23.5
菜品4  商户名称: 黄焖鸡米饭      菜品名: 扬州炒饭          价格: 15.0
请输入对应的菜品序号:
3
菜品收藏成功
```

b.收藏已收藏的菜品

```
请输入指令 (输入 '-h' 查看所有指令):
-fd
请输入想要收藏的菜品名:
炒饭
符合条件的菜品信息如下:
菜品1  商户名称: 阿姨蛋炒饭      菜品名: 梅干菜蛋炒饭      价格: 16.0
菜品2  商户名称: 阿姨蛋炒饭      菜品名: 里脊肉蛋炒饭      价格: 20.0
菜品3  商户名称: 阿姨蛋炒饭      菜品名: 牛肉蛋炒饭        价格: 23.5
菜品4  商户名称: 黄焖鸡米饭      菜品名: 扬州炒饭          价格: 15.0
请输入对应的菜品序号:
1
您已经收藏过此菜品!
```

c.收藏不存在的菜品

```
请输入指令 (输入 '-h' 查看所有指令):
-fd
请输入想要收藏的菜品名:
汤
菜品不存在! 收藏失败
```

(5) -fm 收藏喜欢的商家

1.搜索结果有多个:

请输入指令 (输入 '-h' 查看所有指令):

-fm

请输入想要收藏的商家名称:

饭

符合条件的商家信息如下:

1	商家名称: 黄焖鸡米饭	商家评分: 0.0	服务区域: 政肃路
2	商家名称: 阿姨蛋炒饭	商家评分: 0.0	服务区域: 国定路602号

输入您想收藏的商家序号, 若不需要, 输入0退出

1

商家收藏成功

2.搜索结果只有一个:

请输入指令 (输入 '-h' 查看所有指令):

-fm

请输入想要收藏的商家名称:

阿姨

符合条件的商家信息如下:

商家名称: 阿姨蛋炒饭	商家评分: 0.0	服务区域: 国定路602号
-------------	-----------	---------------

您想收藏该商家吗(y/n)

y

商家收藏成功

(6) -sd 搜索特定菜品信息

顾客可直接搜索系统中的某一款菜品查询详情, 包括名称、价格、分类、原料等等:

```

请输入指令 (输入 '-h' 查看所有指令):
-sd
请输入想要查询详细信息的菜品名:
炒饭
符合条件的菜品信息如下:
菜品1  商户名称: 阿姨蛋炒饭  菜品名: 梅干菜蛋炒饭  价格: 16.0
菜品2  商户名称: 阿姨蛋炒饭  菜品名: 里脊肉蛋炒饭  价格: 20.0
菜品3  商户名称: 阿姨蛋炒饭  菜品名: 牛肉蛋炒饭  价格: 23.5
菜品4  商户名称: 黄焖鸡米饭  菜品名: 扬州炒饭  价格: 15.0
请输入对应的菜品序号:
1
菜品名称:      梅干菜蛋炒饭
菜品价格:      16.0
菜品分类:      主食
菜品描述:      梅干菜拌在蛋炒饭里
菜品原料:      梅干菜, 蛋, 饭
可能的过敏源:  梅干菜, 蛋
营养成分:      膳食纤维, 蛋白质, 淀粉
菜品评分:      2.5
菜品图片:      no image
请问是否查看菜品评论(y/n)
y
还没有人评论

```

在商家页面中的菜品搜索详见后续-sm部分。

(7) -o 预订餐厅和点餐

顾客可以直接输入下单指令，并选择商家与菜品：

```

请输入指令 (输入 '-h' 查看所有指令):
-o
您希望在哪里点餐?
请输入想要点单的商家名称:
饭
符合条件的商家信息如下:
1      商家名称: 黄焖鸡米饭  商家评分: 0.0  服务区域: 政肃路
2      商家名称: 阿姨蛋炒饭  商家评分: 0.0  服务区域: 国定路602号
输入您想点单的商家序号, 若不需要, 输入0退出
2
菜单如下:
1      菜品名: 梅干菜蛋炒饭  价格(元): 16.0  菜品类型: 主食
2      菜品名: 里脊肉蛋炒饭  价格(元): 20.0  菜品类型: 主食
3      菜品名: 牛肉蛋炒饭  价格(元): 23.5  菜品类型: 主食
4      菜品名: 配饭一绝西瓜汁  价格(元): 4.0  菜品类型: 饮品
菜单显示完毕
请输入菜品序号表示选择, 用空格间隔, 退出请输入q:
1 2
请输入以上菜品的数量, 用空格间隔, 退出请输入q:
1 1
点餐成功! 订单号1

```

在商家页面中也能进行下单，详见后续-sm部分。

(8) -so 展示所有与自己相关的订单(越新的订单显示越靠前)

```
请输入指令 (输入 '-h' 查看所有指令):
-o
您希望在哪里点餐?
请输入想要点单的商家名称:
饭
符合条件的商家信息如下:
1      商家名称: 黄焖鸡米饭      商家评分: 0.0      服务区域: 政肃路
2      商家名称: 阿姨蛋炒饭      商家评分: 0.0      服务区域: 国定路602号
输入您想点单的商家序号, 若不需要, 输入0退出
2
菜单如下:
1      菜品名: 梅干菜蛋炒饭      价格(元): 16.0      菜品类型: 主食
2      菜品名: 里脊肉蛋炒饭      价格(元): 20.0      菜品类型: 主食
3      菜品名: 牛肉蛋炒饭        价格(元): 23.5      菜品类型: 主食
4      菜品名: 配饭一绝西瓜汁    价格(元): 4.0      菜品类型: 饮品
菜单显示完毕
请输入菜品序号表示选择, 用空格间隔, 退出请输入q:
1 2
请输入以上菜品的数量, 用空格间隔, 退出请输入q:
1 1
点餐成功! 订单号1
```

(9) -co 评价订单

当商家出餐并通知顾客后（详见商家部分），顾客可以对这一订单进行评价，评价后订单状态变为用户已评价，且顾客能够自行选择是否对商家或是对某几道菜进行评价：

```
-co
1      订单号: 4      订单状态: 用户已收到出餐通知      商家名称: 阿姨蛋炒饭      菜品: 配饭一绝西瓜汁 1份
2      订单号: 2      订单状态: 用户已收到出餐通知      商家名称: 阿姨蛋炒饭      菜品: 里脊肉蛋炒饭 2份
请选择你想评价的订单:
1
请问是否需要评价商家(y/n)
y
请为商家评分(1~5)
4
请写出您对该商家的评价
出餐很快, 推荐
菜品:
1      配饭一绝西瓜汁 1份
请输入菜品序号表示选择, 用空格间隔, 退出请输入q:
1
请为配饭一绝西瓜汁评分(1~5)
5
请写出您对该菜品的评价
冰凉爽口, 夏天喝很舒服
请输入指令 (输入 '-h' 查看所有指令):
-so
1      订单号: 4      订单状态: 用户已评价      商家名称: 阿姨蛋炒饭      菜品: 配饭一绝西瓜汁 1份
2      订单号: 3      订单状态: 待出餐      商家名称: 黄焖鸡米饭      菜品: 扬州炒饭 1份
3      订单号: 2      订单状态: 用户已收到出餐通知      商家名称: 阿姨蛋炒饭      菜品: 里脊肉蛋炒饭 2份
4      订单号: 1      订单状态: 待出餐      商家名称: 阿姨蛋炒饭      菜品: 梅干菜蛋炒饭 1份 里脊肉蛋炒饭 1份
```

```
请输入指令 (输入 '-h' 查看所有指令):
-co
1      订单号: 5      订单状态: 用户已收到出餐通知      商家名称: 阿姨蛋炒饭      菜品: 牛肉蛋炒饭 2份
是否评价该订单(y/n)
y
请问是否需要评价商家(y/n)
y
请为商家评分(1~5)
5
请写出您对商家的评价
满分好评!
菜品:
1      牛肉蛋炒饭      2份
请输入菜品序号表示选择, 用空格间隔, 退出请输入q:
q
退出评价
```

当顾客在完成订单评价后, 顾客给出的评分与评价会更新到表 `merchants_comments` 与 `dishes_comments` 中, 并及时计算商家或菜品的新平均评分, 下图为商家拥有一个4分与一个5分评价时显示的平均评分

```
请输入指令 (输入 '-h' 查看所有指令):
-sm
请输入想要搜索并进入的商家名称:
阿姨
符合条件的商家信息如下:
商家名称: 阿姨蛋炒饭      商家评分: 4.5      服务区域: 国定路602号
您想搜索并进入该商家吗(y/n)
```

(10) -sm 搜索商家

在搜索商家后能够选择进入每个商家的界面并在新界面中选择进一步指令, 直到选择退出该页面:

```
请输入指令 (输入 '-h' 查看所有指令):
-sm
请输入想要搜索并进入的商家名称:
阿姨
符合条件的商家信息如下:
商家名称: 阿姨蛋炒饭    商家评分: 0.0    服务区域: 国定路602号
您想搜索并进入该商家吗(y/n)
y
菜单如下:
1      菜品名: 梅干菜蛋炒饭    价格(元): 16.0    菜品类型: 主食
2      菜品名: 里脊肉蛋炒饭    价格(元): 20.0    菜品类型: 主食
3      菜品名: 牛肉蛋炒饭    价格(元): 23.5    菜品类型: 主食
4      菜品名: 配饭一绝西瓜汁    价格(元): 4.0    菜品类型: 饮品
菜单显示完毕
可选指令如下:
1 显示菜单详情
2 下单
3 查看商家评价
4 收藏该商家
5 查询菜品详细信息
6 收藏某一菜品
7 查看商家主打菜
8 显示指令列表
9 退出该商家
请选择需要进行的操作(输入8查看所有指令):
```

下面依次展示商家界面中的各功能效果:

a.显示菜单详情

```
请选择需要进行的操作(输入8查看所有指令):
1
菜单如下:
1      菜品名: 梅干菜蛋炒饭    价格(元): 16.0    菜品类型: 主食
2      菜品名: 里脊肉蛋炒饭    价格(元): 20.0    菜品类型: 主食
3      菜品名: 牛肉蛋炒饭    价格(元): 23.5    菜品类型: 主食
4      菜品名: 配饭一绝西瓜汁    价格(元): 4.0    菜品类型: 饮品
菜单显示完毕
```

b.下单

请选择需要进行的操作(输入8查看所有指令):

2

菜单如下:

1	菜品名: 梅干菜蛋炒饭	价格(元): 16.0	菜品类型: 主食
2	菜品名: 里脊肉蛋炒饭	价格(元): 20.0	菜品类型: 主食
3	菜品名: 牛肉蛋炒饭	价格(元): 23.5	菜品类型: 主食
4	菜品名: 配饭一绝西瓜汁	价格(元): 4.0	菜品类型: 饮品

菜单显示完毕

请输入菜品序号表示选择, 用空格间隔, 退出请输入q:

2

请输入以上菜品的数量, 用空格间隔, 退出请输入q:

2

点餐成功! 订单号2

c.查看商家评价

请选择需要进行的操作(输入8查看所有指令):

3

评论人: HCY 评论内容: 出餐很快, 推荐

评论人: Cxy 评论内容: 满分好评!

d.收藏该商家

请选择需要进行的操作(输入8查看所有指令):

4

您已经收藏过此商家!

e.查询菜品详情

```
请选择需要进行的操作(输入8查看所有指令):
5
请输入想要查询详细信息的菜品名:
西瓜汁
符合条件的菜品信息如下:
菜品名: 配饭一绝西瓜汁  价格: 4.0
您想查询详细信息该菜品吗(y/n)
y
菜品名称:      配饭一绝西瓜汁
菜品价格:      4.0
菜品分类:      饮品
菜品描述:      5
菜品原料:      5
可能的过敏源:  5
营养成分:      5
菜品评分:      5.0
菜品图片:      no image
请问是否查看菜品评论(y/n)
y
评论人: HCY      评论内容: 冰凉爽口, 夏天喝很舒服
```

f.收藏某一菜品

```
请选择需要进行的操作(输入8查看所有指令):
6
请输入想要收藏的菜品名:
西瓜汁
符合条件的菜品信息如下:
菜品名: 配饭一绝西瓜汁  价格: 4.0
您想收藏该菜品吗(y/n)
y
菜品收藏成功
```

注：当在商家页面中若收藏别的商家提供的菜，则无法成功（扬州炒饭为店铺“黄焖鸡米饭”中的菜而当前位于“阿姨蛋炒饭”店铺界面），如下图：

```
请选择需要进行的操作(输入8查看所有指令):
6
请输入想要收藏的菜品名:
扬州炒饭
菜品不存在！收藏失败
```


g.查看该商家招牌菜

请选择需要进行的操作(输入8查看所有指令):

7

菜品名: 梅干菜蛋炒饭	价格: 16.0	菜品类型: 主食
菜品名: 牛肉蛋炒饭	价格: 23.5	菜品类型: 主食

h.显示指令列表

请选择需要进行的操作(输入8查看所有指令):

8

可选指令如下:

- 1 显示菜单详情
- 2 下单
- 3 查看商家评价
- 4 收藏该商家
- 5 查询菜品详细信息
- 6 收藏某一菜品
- 7 查看商家主打菜
- 8 显示指令列表
- 9 退出该商家

i.退出商家

请选择需要进行的操作(输入8查看所有指令):

9

请输入指令 (输入 '-h' 查看所有指令):

-q、-exit结果与上同，此处省略

六、核心SQL语句说明

由于我们使用的SQL语句并不复杂，以下仅挑选几个例子讲解。

以下这个方法用于在特定表格中挑选出符合条件的元组，如isInfoClear==false，使用的是

```
query = "SELECT * FROM " + tableName + " WHERE " + infoName + " like '%" + info + "%' ";
```

对应了我们实现的模糊搜索的功能。

```

        public static ResultSet isItemExists(Connection connection,
String tableName, String info, String infoName,
        boolean isInfoClear) throws SQLException {
    String query;
    // 在创建 PreparedStatement 时指定可以滚动的 ResultSet 类型
    PreparedStatement preparedStatement;
    if (isInfoClear) {
        query = "SELECT * FROM " + tableName +
            " WHERE " + infoName + " = ?";
        preparedStatement = connection.prepareStatement(query,
ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
        preparedStatement.setString(1, info);
    } else {
        query = "SELECT * FROM " + tableName +
            " WHERE " + infoName + " like '%" + info + "%'
";
        preparedStatement = connection.prepareStatement(query,
ResultSet.TYPE_SCROLL_INSENSITIVE,
            ResultSet.CONCUR_READ_ONLY);
    }
    return preparedStatement.executeQuery();
}

```

以下这个方法被Merchant调用，用于添加菜品。在这个方法中，我们使用了事务的特性，若获取自增菜品ID的过程失败，添加菜品的操作也会被回滚，这保证了数据库数据的一致性（在Customer点餐的过程中，我们采用了类似的操作）：

```

public static int insertDish(Connection connection, Object[] data)
{
    String query = "INSERT INTO Dishes (merchant_id, name,
price, type, description, " +
        "ingredients, possible_allergens, nutrition,
rating, image) " +
        "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
    try {
        // 开始事务
        connection.setAutoCommit(false);
        PreparedStatement pstmt =
connection.prepareStatement(query,
Statement.RETURN_GENERATED_KEYS);

```

```

        for (int i = 0; i < data.length; i++) {
            pstmt.setObject(i + 1, data[i]);
        }
        pstmt.executeUpdate();
        ResultSet generatedKeys = pstmt.getGeneratedKeys();
        if (generatedKeys.next()) {
            return generatedKeys.getInt(1);
        } else {
            throw new SQLException("无法获取菜品ID");
        }
    } catch (Exception e) {
        System.out.println("向数据库中添加菜品时出错 " +
e.getMessage());
        // 语句执行异常,在catch中进行回滚事务
        try {
            connection.rollback();
        } catch (SQLException ex) {
            System.err.println("事务回滚出错");
        }
    } finally {
        // 在关闭资源之前进行提交,无论是否执行回滚,都需要进行提交,所以将提交放在finally里面
        try {
            connection.commit();
            connection.setAutoCommit(true);
        } catch (SQLException e) {
            System.out.println("添加菜品时事务提交出错");
        }
    }
    return -1;
}

```

以下是一个插入数据的通用方法，使用了PreparedStatement 执行插入操作，防止了SQL注入，由于Object[] data在外部已经被检测过，此处不再另做检测：

```

public static void insertData(Connection connection, String
tableName, Object[] data) throws SQLException {
    StringBuilder insertQuery = new StringBuilder("INSERT INTO
");
    insertQuery.append(tableName).append(" VALUES (");

```

```

// 构建插入数据的占位符
for (int i = 0; i < data.length; i++) {
    insertQuery.append("?");
    if (i < data.length - 1) {
        insertQuery.append(",");
    }
}
insertQuery.append(")");

// 使用 PreparedStatement 执行插入操作
try (PreparedStatement pstmt =
connection.prepareStatement(insertQuery.toString())) {
    // 设置插入数据的值
    for (int i = 0; i < data.length; i++) {
        pstmt.setObject(i + 1, data[i]);
    }
    // 执行插入操作
    pstmt.executeUpdate();
}
}

```

七、运行方法

由于我们采取手动添加而不是从文件读取数据的方式，无法提供数据库的数据，但是我们提交了可以运行的源代码，可以根据上面功能实现的提示进行操作，验证结果。