

实现の功能

实现の功能

题干

数据库Schema

1.实现登录

(1)用户认证

(2)用户注册

2.consumer

(1)获取历史订单

(2)订餐

(3)投诉--不实现

3.delievery

(1)选择送餐单

(2)改变送餐状态

4.manager

(1)用户注册

(2)管理菜品--基础的增删改查

(3)查看每日的销售情况

(4)简单的数据统计

H2 题干

作为顾客，可以查看菜单、订餐（如果未登录，需先登录）、支付、投诉等。

餐厅员工可以进行接受订单、生成付费请求、备餐和请求送餐。

餐厅经理是餐厅管理层，可以管理菜单、查看每日销售情况及简单数据统计、投诉。

送餐员可以选择确认送餐单、确认送餐到达时间、确认已送到。

顾客订餐过程如下：

1. 顾客请求查看系统显示的菜单；
2. 顾客选菜；

3. 系统显示订单和价格及可送餐时间；
4. 顾客确认订单，指定送餐时间、地点并支付；
5. 系统确认接受订单，给顾客的界面可显示订单状态（确认中、备餐中、送餐中及预计到达时间），同时发送相关订餐信息通知给餐厅员工。

H2 数据库Schema

11.10更新：加入联系集：consumer_order

11.9更新：order加入时间属性：YY:MM:HH MM:SS酱什的

```
1 create database takeaway;# 创建数据库
2
3 create table user(# 将所有的员工全部集合在了这一个表中
4     type tinyint comment '1=manager,2=delivery,3=consumer',
5     ID char(20),
6     password char(20),
7     addID char(20),
8     foreign key (addID)references address(ID),
9     constraint checkaddr check ( type=3&addID is not null )
10 );
11
12 create table dish(
13     type char(10) comment '菜的类型，是甜品或是蛋糕之类',
14     name char(10) comment '菜名',
15     ID char(20) primary key auto_increment,
16     money int comment '价格',
```

```
17     introduction char(50) comment '简短的介绍'
18 );
19
20 create table address(
21     ID char(20) primary key ,
22     phonenum char(11),
23     building char(5)comment '哪栋楼',
24     floor tinyint comment '哪层'
25 );# 对于phonenum要设置为有且仅有11位，从java程序上写。
26
27 create table order(
28     ID char(20),
29     addID char(20),
30     dishID char(20),
31     dishAmount tinyint,
32     money int,
33     state tinyint,
34     deliveryID char(20),
35     orderTime datetime,
36     foreign key (addID)references address(ID),
37     foreign key (dishID)references dish(ID)
38 );
39
40 #联系集
41 create table consumer_order(
42     consumerID char(20),
43     orderID char(20),
44     foreign key (consumerID)references user(ID),
45     foreign key (orderID) references order(ID)
46 );
```

H2 1.实现登录

简化下，不对用户的权限进行设置。

限制用户的权限从java程序设置，而不是在数据库设置。

H3 (1)用户认证

描述：首先进入登录界面——

设置有三项信息(雾)，向后端传递三个信息： `type, ID, pw`

后端获取这三个信息后，去向数据库查询该 `user` 是否存在，如果存在就进入用户的界面。

H3 (2)用户注册

描述：相当于是从manager这去注册信息，然后用户可以使用。有且仅有从manager这注册一种方式。

```
manager insert usr(type, ID, pw)
```

H2 2.consumer

H3 (1)获取历史订单

发现一个很离谱的：就是用户登录上去的同时，应该获取到其所有的历史订单，这个要先查，然后向用户展示。

描述：在用户登上去的瞬间通过其ID查找 `consumer_order` 的表找到相关的订单ID。

简化，不显示具体每一单都做了什么，只显示ID号。列完所有的订单ID，做跳转的按钮，按了之后可以显示订单的详细信息。

实现：根据订单号查找 **order** 的详细信息并向前端传递。

至于显示的订单信息页面，自行设计。

```
1 create table consumer_order(  
2     consumerID char(20),  
3     orderID char(20),  
4     foreign key (consumerID) references user(ID),  
5     foreign key (orderID) references order(ID)  
6 );
```

H3 (2)订餐

菜品展示：点击某菜系可以获取到该菜系每一个菜品的详细信息

相当于是查询type="菜系"的所有dish

菜品点击：点击可以计算金额

地址固定，不需要进行选择。在manager创建时会给出相应的地址。

支付：需要使用到微信支付功能，该功能需要小程序企业版才能使用。就在此不进行实现，但是在写报告的时候要写出是版本不一样才不能实现，并给出实现相应的方法。

H3 (3)投诉--不实现

不太会实现，先把基础概念写完再讨论看看(如果有时间的话)。

大概的想法就是：用户键入的长字符串将会存入数据库中，并将该字符串发送给manager。

manager一登陆，就会执行check函数：看是否有未读的投诉信息；如果有，就提醒manager去查看。

H2 3.delievery

H3 (1)选择送餐单

对于订单状态是0(没有骑手接单的状态)，其delieveryID=null。将这类的订单的信息推送到delievery的页面上，当delievery选择后改变该订单的状态；

但是注意对于多名骑手：一定注意信息的同时性，不能存在两人接一单的情况。

每一时刻，如果对于数据库的数据进行了更新，就要同步更新所有给出的信息。

H3 (2)改变送餐状态

选择：

- 0——等待骑手接单
- 1——骑手已接单
- 2——骑手已取到餐，正在送餐中
- 3——骑手已送达

H2 4.manager

H3 (1)用户注册

首先现在address中先创建地址，然后再insert到usr表中详细信息。

H3 (2)管理菜品--基础的增删改查

首先将所有的菜品的信息显示出来，然后给按钮点击选择增删改。

H3 (3)查看每日的销售情况

首先获取今天的信息，然后查询时间是今天的订单并显示。

H3 (4)简单的数据统计

这块就将每天的订单进行简单的统计，获取[每天]:

1. 不同菜系的订单数
2. 不同菜系的金额
3. 所有订单的总金额

就这三吧，然后想想怎么画图。