华南理工大学

国际校区点餐系统

程序设计书

课程名称：	JAVA 程序设计
学生姓名：	姚慧琳
学生学号：	202164700523
学生专业：	数据科学与大数据技术
开课学期：	2022-2023 学年 第一学期

2023 年 1 月

目录

一 绪论

1.1 课题目标

本系统致力于设计一个能够保证信息准确性和时效性，随时对餐饮信息进行管理的餐厅点餐管理信息平台，使得餐厅点餐各项信息快速准确传达给顾客和餐厅人员，为餐厅点餐信息交流提供便利。

1.2 课题背景

传统的餐饮行业大多通过员工手写下单记录大量点餐信息，工作量大，且容错率较低。同时，不断变化的餐饮信息与繁杂的账单计算都需要耗费员工大量的时间与脑力。

如今，通过程序设计餐厅点餐系统则能够很好的减少这一弊端。后厨能够通过系统更新菜品信息与余量，消费者仅需通过系统便可实现一键下单与结账。这极大的减少了双方的时间与精力，同时增加了消费者对餐饮行业的满意程度。

1.3 课题意义

通过使用本系统，一方面能够简化点餐环节，顾客仅需通过本系统即可实时获得所有餐品信息进行下单，极大的提高了顾客满意度，加快了顾客的点餐效率，另一方面通过系统点餐能够减低餐厅员工的劳动强度，并提高餐厅管理的透明化与规范化程度，于买卖双方均是一个百利而无一害的选择。

二 系统概述

2.1 程序总功能

本系统以国际校区韩式餐厅为例，依托 IDEA 平台与 JAVA 语言搭建餐厅点餐系统。该系统共分为管理端与用户端两大模块。用户可通过个人用户端进入点餐平台进行操作，平台提供到店点餐、查看历史账单、查看个人信息等服务。新用户还可通过注册界面注册个人用户。管理人员可通过联系平台维护人员进行菜品修改、查询订单、增加管理人员信息等操作。

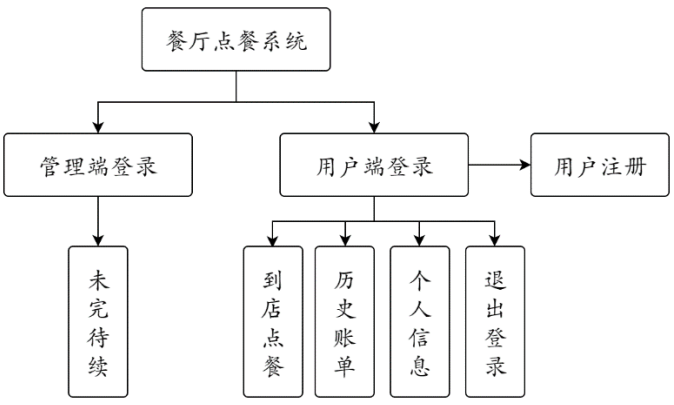

图 1 程序总功能模块图

2.2 程序总流程

客户、商家通过本餐厅点餐系统进行信息传输，客户向系统提交订单与个人信息，商家向系统提交菜品信息，同时接收系统给出的订单信息。而系统所接收的所有数据均发送并存储到数据库当中，系统所发送的数据也均从数据库中提取。

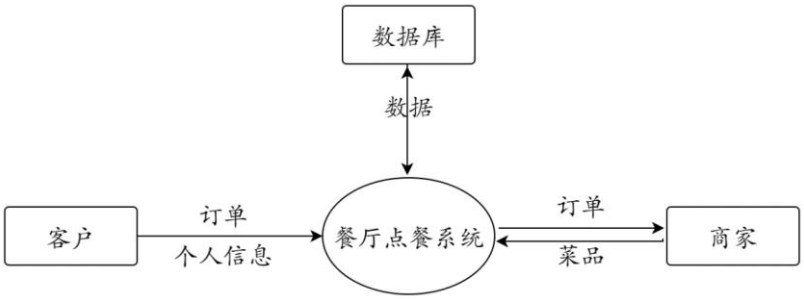

图 2 程序总流程模块图

2.3 程序文件设计结构

如下图所示，文件共分成了八大模块。其中，4 模块是构建 JDBUtil 类通过 JDBC 接口实现与数据库的连接与断开。2 模块是根据数据库中的四张表格（`food`、`Order_food`、`Order_info`、`user`）创建的四个大类（Food，Order_food，Order_info，User），用于创建临时存储对象用于程序与数据库之间的信息传输。3 模块是构建对数据库进行增删改查的方法，通过 2 与 4 模块的信息发送或提取数据库对应内容。7 模块用于搭建系统可视化界面，并通过 3 模块获取与传输数据库信息。5 模块用于存放各类小工具函数。1 与 6 主要用于存放各类照片，便于系统直接展示与调用。

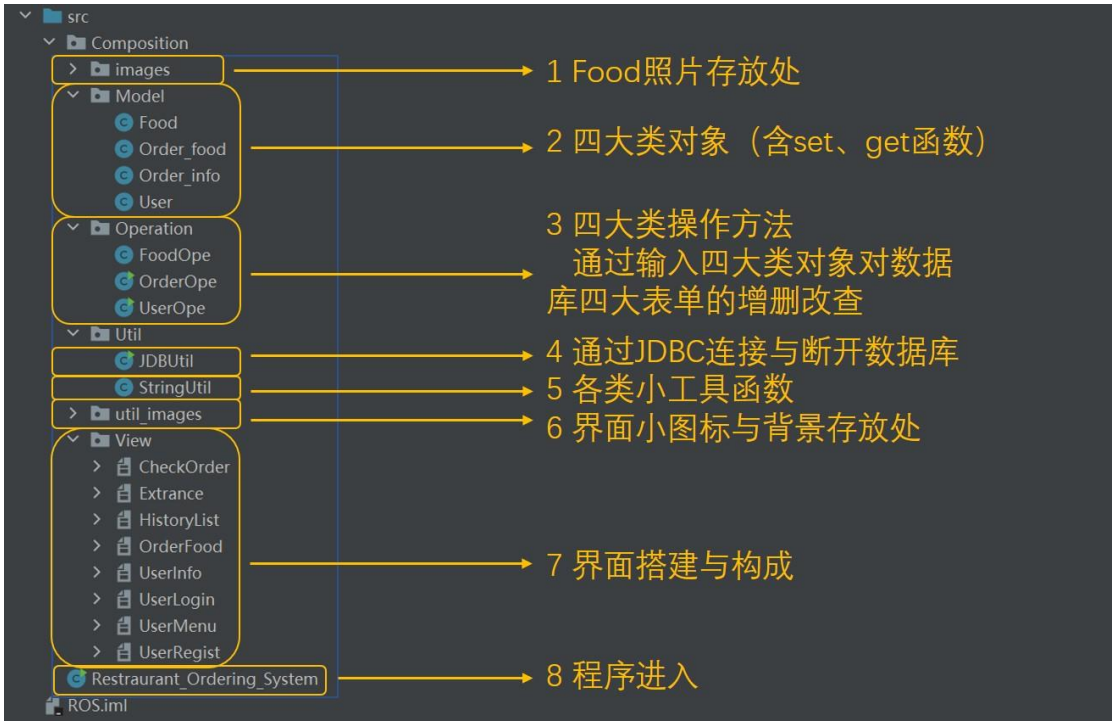

图 3 程序文件设计结构图

2.4 开发工具

（开发环境）JDK 11.0.12.7+（运行平台）Window 11+（数据库）MySQL 5.7

三 程序详细设计

3.1 数据库存储

程序通过 IDEA 插件 Database 成功连接上 mysql 库，并根据需求创建`food`、`Order_food`、`Order_info`、`user`四个表单。

Table: Food

数据项名	数据含义	数据类型
Food_id(key)	食物序号	int(2)
Food_name	食物名称	varchar(30)
Food_type	食物类型	varchar(10)
Food_price	食物单价	float
Food_image	图片链接	varchar(100)
Food_detail	食物介绍	varchar(100)

Table: Order_food

数据项名	数据含义	数据类型
Index(key)	序号	int
Order_id	订单号	int(4)
Food_id	食物序号	varchar(20)
Food_price	食物单价	float
Food_amount	食物数量	int(2)
Food_sum_price	食物总价	float

Table: Order_info

数据项名	数据含义	数据类型
Order_id(key)	订单号	int(4)
Order_table	桌号	int(4)
Order_status	订单状态	int(1)
Order_Amount	餐品数量	int(4)
Order_debt	餐品总价	float
User_id	客户账号名	varchar(10)
Order_date	下单日期	varchar(10)
Order_time	下单时间	varchar(10)

Table: User

数据项名	数据含义	数据类型
User_id	客户账号名	varchar(10)
User_pwd	客户密码	int(8)
User_mail	客户邮箱	varchar(20)
User_phone	客户电话	varchar(20)

(1) Food：存储餐品基本信息，每件餐品对应一行数据，仅餐厅员工可对餐品信息进行增删改除操作。

(2) Order_Info：存储订单基本信息，每个订单对应一行数据，客户和员工均可对订单信息进行条件查询，但无法更改。

(3) Order_food：存储订单菜品信息，每个订单的每个菜品对应一行数据，客户和员工均可对订单菜品信息进行条件查询，但无法更改。

(4) User：存储客户个人信息，每个客户对应一行数据，客户可对个人信息进行增删改操作，但用户名不能与其他人重复。

3.2 程序与数据库的连接

程序通过导入 jar 包 mysql-connector-java 后成功通过 JDBC 接口规范连接至本 JAVA 程序，随后调用 JDBC 中的 PreparedStatement 接口用于对数据库的增删改查。

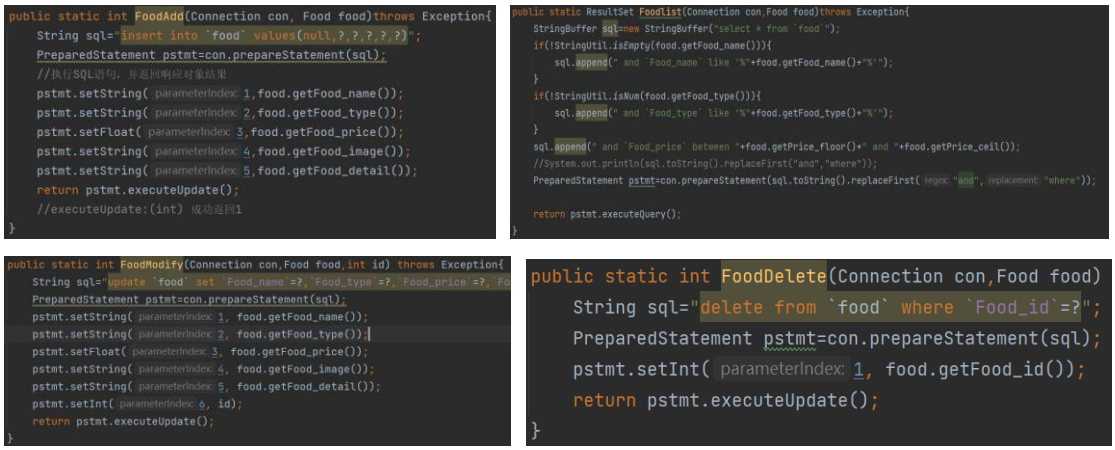

图 4 以`Food`为例的增删改查

3.3 界面设计

本程序的界面设计均是通过 IDEA 自带的 SWING UI DESIGNER 进行制作，通过绘制 FORM 文件与添加各个界面的相关代码组合而成。

3.3.1 界面·到店点餐

界面整体使用 BorderLayout 布局，将版面分为东南西北中五大块。

北部区域为标题部分，通过设置 JPANEL 使其居中与变大。

西部区域为菜式导航部分，可通过点击不同菜式按钮 JButton，实现中部区域餐饮界面的切换。

图 5 到店点餐界面展示

中部区域为菜品展示区，该区域构建时通过实时读取数据库中的`Food`数据以构建对应的 JPanel 并将其加入到对应的菜式 JPanel 当中。因此，餐厅工作人员可以通过特定界面或联系工程师进行餐品的定期维护、修改与添加。

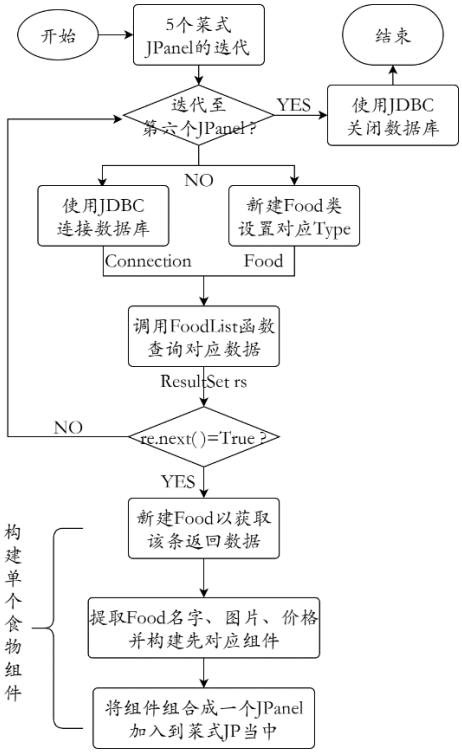

图 6 中部界面构建流程图

```java
//init：原始界面更新(main)
5 usages
private void menu_get(JPanel jp,String type) throws Exception {
    Food food=new Food();
    food.setFood_type(type);

    menu_view(jp,food);
}
//菜单界面刷新与创建(utility)
3 usages
private void menu_view(JPanel jp,Food food) throws Exception {
    jp.removeAll();
    jp.setLayout(new GridLayout( rows: 0, cols: 3));

    JDBUtil jdbUtil=new JDBUtil();
    //进行改写
    Connection con=null;
    try{
        con= jdbUtil.getCon();
        ResultSet rs= FoodOpe.Foodlist(con,food);
        while(rs.next()){...}//界面构建
    } catch (Exception e) {
        throw new RuntimeException(e);
    }finally {
        jdbUtil.closeCon(con);
    }

    jp.updateUI();
}
```

图 7 中部界面构建代码（部分隐藏）

西部区域共分成两大块。一块是通过选择填写菜品名称、菜品类型、价格区间条件筛选处符合条件的菜品。该区域先是通过获取筛选 JTEXT 的值，从数据库中获取符合条件的食物，再展示到中部面板当中。具体原理与中部菜品展示相似，但再构建 FOOD 类中仍还费了一定心思。

另外一块是按钮栏。“添加”按钮用于将选中商品加入到西部面板的购物车当中。点击该按钮后，其能够通过扫描 JPanel 中各 JCheckBox 是否被选中来判断哪些食物需要被加入购物车中。若为第一次添加，则自动显示添加茶味费一栏。若该商品已在购物车中，则数量加倍；若不在，则添加至购物车中。购物车由 JSrollPane 内嵌 JTable 组成，将数据添加到 Vector 后便可作为行数据直接添加至购物车中。“删除”按钮亦是类似原理，将选中商品从购物车中删除。

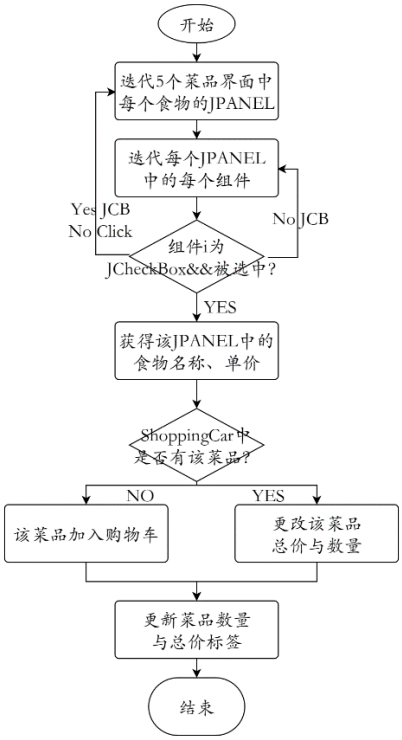

图 8 添加购物车构建流程图

```
private void add_ActionListener(ActionEvent e){
    DefaultTableModel dtm=(DefaultTableModel)shoppingcar.getModel();
    if(!is_select_DeskandAmount(dtm)) return;
    JPanel[]jp={menu1,menu2,menu3,menu4,menu5};
    for(int i=0;i<5;++i){
        int co=jp[i].getComponentCount();
        for(int j=0;j<co;++j){
            //0 name:checkbox 1 price:jlabel 2:jpanel 3:image
            JPanel J=(JPanel)jp[i].getComponent(j);
            JPanel K=(JPanel)J.getComponent( n: 0);
            //0:CHECKBOX 1:LABEL
            JCheckBox jcb=(JCheckBox) K.getComponent( n: 0);
            JLabel jla=(JLabel) K.getComponent( n: 1);
            if(jcb.isSelected()){
                int r=in_shoppingcar(dtm,jcb.getText());
                if(r!=-1){
                    //物品在购物车，数量+1即可
                    jcb.setSelected(false);
                    int num=(int)dtm.getValueAt(r, column: 2)+1;

                    String p1=jla.getText().substring(0,jla.getText().length()-1);
                    String p2=(String) dtm.getValueAt(r, column: 4);//以前商品的总价
                    float p=Float.parseFloat(p2.substring(0,p2.length()-1))+Float.
                    String price=StringUtil.FtoY(p);
                    dtm.setValueAt(num,r, column: 2);
                    dtm.setValueAt(price,r, column: 4);
                }else{
                    //物品不在购物车中:直接添加
                    jcb.setSelected(false);
                    Vector row=new Vector();
                    int n=dtm.getRowCount();
                    row.add(n+1);
                    row.add(jcb.getText());//1名称
                    row.add(1);//2数量
                    row.add(jla.getText());//3单价
                    row.add(jla.getText());//4总价
                    dtm.addRow(row);
```

图 9 添加购物车构建代码（部分隐藏）

“下单”按钮用于将购物车中菜品以订单形式传输给后厨，并展示到南部区域的订单详情界面。当点击该按钮后，其直接获取其 shoppingcar 中的数据，并将其逐条导入数据库中，同时删除 shoppingcar 相关数据，并添加订单详情中的相关数据。“加单”按钮功能与“下单”按钮相似，但必须先在该页面点击“下单”按钮后才可启用“加单”按钮。

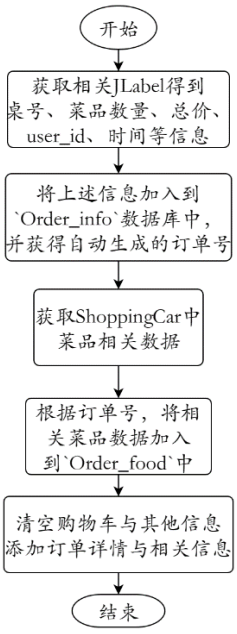

图 10 下单按钮构建流程图

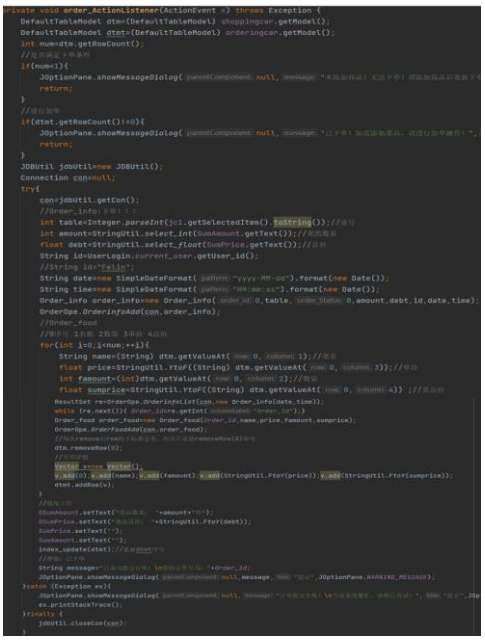

图 11 下单按钮构建代码

“结账”按钮用于修改订单“未支付”状态，点击后将弹窗界面并跳转至订单详情界面。

3.3.2 界面·历史账单

该界面用于显示该用户过去在这家餐厅的全部消费记录。界面使用了 JLayerPane 将背景与主布局置于 DEFAULT_LAYER 和 MODEL_LAYER 层之中以设置背景图片。同时，主界面构建是通过具体订单数据合理使用 GridBagLayout 布局以将各个组件整齐排列在同一个 JPanel 中，再将其加入到 JScollpane 中实现滚动查看相关订单的效果。

```
//第二行，菜单
ResultSet re=OrderOpe.OrderfoodList(con,new Order_food(oi.getOrder_id()));
while(re.next()){
    Order_food of=new Order_food(re.getInt( columnLabel: "Order_id"),re.getSt

    ResultSet rs= FoodOpe.Foodlist(con,new Food(of.getFood_name()));
    while(rs.next()) {
        Food f=new Food(rs.getString( columnLabel: "Food_name"),rs.getString(
        bgc.gridwidth=5;
        bgc.gridheight=5;
        JLabel image=new JLabel();
        image.setIcon(itran(f.getFood_image(), w: 60, h: 50));
        gbl.setConstraints(image,bgc);
        jp.add(image);
    }

    bgc.gridwidth=10;
    bgc.gridheight=5;
    JLabel name=new JLabel();
    name.setText(of.getFood_name());
    name.setHorizontalAlignment(0);
    name.setVerticalAlignment(1);
    gbl.setConstraints(name,bgc);
    jp.add(name);//一定记得把组件加进去

    bgc.gridwidth=GridBagConstraints.REMAINDER;
    bgc.gridheight=5;
    JLabel amount=new JLabel();
    amount.setText("x"+of.getFood_amount());
    amount.setHorizontalAlignment(SwingConstants.CENTER);
    amount.setVerticalAlignment(1);
    gbl.setConstraints(amount,bgc);
    jp.add(amount);
}
```

图 12 使用 GridBagLayout 布局构建组件

```
//设置背景
layeredPane=new JLayeredPane();
JPanel background=new JPanel();
background.setBounds( x: 0, y: 0, width: 750, height: 500);
JLabel bg=new JLabel(itran( name: "src/Composition/util_image
background.add(bg);
layeredPane.add(background,JLayeredPane.DEFAULT_LAYER);

//设置返回键
JLabel back=new JLabel();
back.setIcon(itran( name: "src/Composition/util_images/back.j
back.setBounds( x: 520, y: 350, width: 60, height: 30);
back.setBackground(Color.WHITE);
layeredPane.add(back,JLayeredPane.POPUP_LAYER);

back.addMouseListener((MouseAdapter) mouseClicked(e) -> {
        super.mouseClicked(e);
        dispose();
        new UserMenu().setVisible(true);
});

//设置panel
Panel.setBackground(new Color( r: 187, g: 187, b: 187, a: 200));
Panel.setBounds( x: 180, y: 100, width: 400, height: 250);//panel
layeredPane.add(Panel,JLayeredPane.MODAL_LAYER);
panel.setBackground(new Color( r: 255, g: 255, b: 255, a: 0));
```

图 13 使用 JLayerPane 构建 JFrame 层级关系代码

3.4 安全保护

本程序在多处可能存在程序崩溃处设置了安全弹窗保护机制，并通过 try 与 catch 方法对崩溃段落进行保护，极大提高了程序的容错率。

图 14 各种消息弹窗提醒

四 程序测试

4.1 界面·点餐入口

程序入口为点餐入口，显示完好！点击“用户入口”即可进入“用户登录”界面。

图 15 入口界面

4.2 界面·用户登录

在本界面，点击“注册”按钮会跳转至“用户注册”界面进行注册。点击“清空”按钮，会将账号栏与密码栏统一清空。输入对应账号和密码后，点击“登录”按钮，显示结果如下：

图 16 正确用户名与用户密码

图 17 输入不存在用户名

图 18 输入错误密码

4.3 界面·用户注册

在本界面，灰色字体为框内提示部分，在框内按下鼠标即可消除灰色字体。点击“返回”按钮即可返回“用户登录”界面；点击“清空”按钮，即可清空填空栏的所有信息。同时，

输入相关内容，点击“确认注册”，显示结果如下：

可以发现，成功注册后 `User` 数据库已经成功显示新注册用户。

图 19 初始界面（灰色字体）

图 20 填写后界面（黑色字体）

	User_id	User_pwd	User_mail	User_phone
1	FELIN	123	2645691822@qq.com	13829292366
2	oy	132	3847@qq.com	13829384938

图 21 `User`数据库实时显示

错误信息提示：

图 22 账号重复

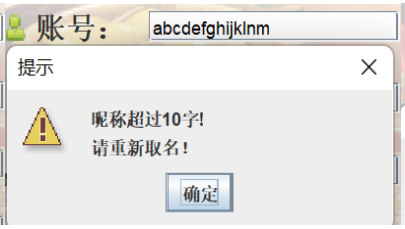

图 23 账号名过长

图 24 密码不一致

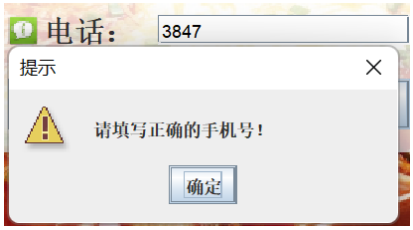

图 25 未填写正确手机号

图 26 未填写正确邮箱

4.4 界面·用户界面菜单

成功“登录”后将会进入该界面，用户可根据自己的需求进入特定页面中。点击“到店点餐”按钮，将会跳转至“到店点餐”界面；点击“历史账单”按钮，将会跳转至“历史账单”界面；点击“个人信息”按钮，将会调转至“个人信息”界面；点击“退出登录”按钮，将会跳转至“用户登录”界面。

图 27 用户界面菜单

4.5 界面·到店点餐

4.5.1 界面展示

点击“到店点餐”按钮，进入该界面。该界面分成了五大块区域：北部区域是标题；中部是餐品面板展示部分；西部区域是菜品导航，可通过点击按钮进入特定餐饮类别；南部区域上半部分是菜品筛选区域，可通过输入特定条件筛选菜品；南部区域下半部分是功能区域，拥有“添加菜品至购物车”、“删除购物车菜品”、“下单”、“加单”、“结账”等功能；东部区域是订单显示区域，用户可在此选择对应桌号与用餐人数，同时下方是购物车待定区域与订单详情展示区域。

图 28 到店点餐界面展示

4.5.2 餐品筛选

筛选 1：菜品名称：辛拉面；类型：主食类；价格下限：20；价格上限：50
筛选结果：肥牛辛拉面、泡菜辛拉面
筛选 2：菜品名称：无；类型：包饭类；价格下限：10；价格上限：无
筛选结果：（如图所示）包饭类价格高于等于 10 的菜品

图 29 筛选菜品截图 1

图 30 筛选截图 2

4.5.3 添加&删除购物车

例子：

“添加”：通过两次点击“添加”按钮增加“泡菜辛拉面”*2、“泡菜汤拉面”*2、“鸡肉石锅拌饭”*1、“豆腐脑汤泡饭”*1、“明太鱼汤泡饭”*2、“五花肉炒饭”*2。由东部区域观察可以发现，此次订单桌号为 02，用餐人数为 4 人，菜品数量为 14 件，总价共计 331.90 元，菜品信息从购物车中可以详细获取。

“删除”：通过 1 次点击“删除”按钮删除“豆腐脑汤泡饭”与“五花肉炒饭”可以发现，原来为 1 份的“豆腐脑汤泡饭”数据行已经被彻底删除，原来为 2 份的“五花肉炒饭”数量从 2 份变至 1 份，同时菜品数量降为 12 件，总价更改为 310.30 元。

图 31 添加购物车

图 32 删除购物车

错误信息提示：

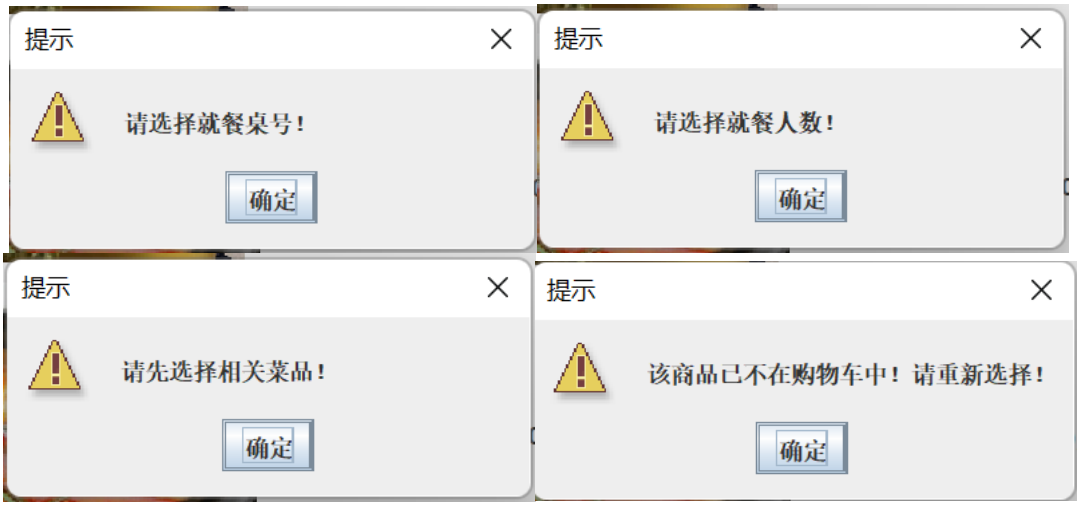

图 33 相关报错提醒

4.5.4 下单&加单

“下单”：在购物车有商品的时候点击“下单”按钮，即可将购物车中内容成功下单并

展示在订单详情当中。

“加单”：在下单后再次添加菜品到购物车后点击“加单”按钮，即可成功加单。同时，通过观察可知，下图加单了“五花肉炒饭”、“参鸡汤泡饭”、“韩国炸酱饭”。对于已下单的菜品可直接添加数量，对于未下单的菜品直接添加行来获得。

通过对 `Order_info` && `Order_food` 数据库的实时提取，我们可以发现 73 号账单与菜品已经添加到其中，且此时 73 号账单状态为 0（未结账）状态.

图 33 “下单”程序截图

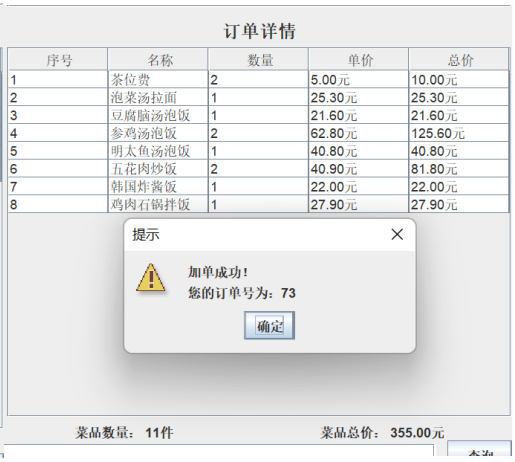

图 34 “加单”程序截图

	Order_id	Order_table	Order_status	Order_Amount	Order_debt	User_id	Order_date	Order_time
1	67	0000000003	1	5	126	Felin	2023-01-03	05:38:21
2	68	0000000003	1	4	63.2	Felin	2023-01-03	06:53:45
3	69	0000000002	1	2	10	FELIN	2023-01-05	02:44:55
4	70	0000000003	1	2	10	FELIN	2023-01-05	02:48:52
5	71	0000000002	1	12	310.3	FELIN	2023-01-05	04:28:39
6	72	0000000003	1	0	0	FELIN	2023-01-05	04:31:33
7	73	0000000002	0	11	355	FELIN	2023-01-05	04:32:17

图 35 `Order_Info`数据库实时信息

	Order_id	Food_name	Food_price	Food_amount	Food_sum_price	index
7	68	泡菜汤拉面	25.3	1	25.3	101
8	69	茶位费	5	2	10	102
9	70	茶位费	5	2	10	103
10	71	茶位费	5	4	20	104
11	72	茶位费	5	2	10	105
12	73	茶位费	5	2	10	106
13	73	泡菜汤拉面	25.3	1	25.3	107
14	73	豆腐脑汤泡饭	21.6	1	21.6	108
15	73	参鸡汤泡饭	62.8	2	125.6	109
16	73	明太鱼汤泡饭	40.8	1	40.8	110
17	73	五花肉炒饭	40.9	2	81.8	111
18	73	韩国炸酱饭	22	1	22	112
19	73	鸡肉石锅拌饭	27.9	1	27.9	113

图 36 `Order_Food`数据库实时更新

错误信息提示：

图 37 相关报错提示

4.5.5 结账

点击“结账”按钮便跳出弹窗再次确认是否结账，选择“是”后，数据库将该订单状态设为 1（已结账），并弹出弹窗告知结账结果。在关闭弹窗后，系统自动跳转至下一界面“订单详情”界面，此处详细展示已结账的菜品信息。点击该界面的“返回”按钮后，返回“用户界面菜单”。

图 38 “结账”程序截图

4.6 界面·历史账单

该界面用于展示该用户 ID 在过去的账单记录，并清晰展示其订单号、订单状态、菜品信息、时间、价格、数量等信息。同时，点击“返回”按钮即可返回“用户菜单界面”。

图 39 历史账单界面

4.7 界面·个人信息

该界面用于展示用户个人信息，同时可通过“编辑”与“保存”按钮成功修改。

图 40 个人信息界面

五 小结

该程序使用 Swing GUI 制作了可视化界面，且通过 JDBC 连接数据库进行数据存储，实现了大容量存储与获取实时信息的两大优点。但同时，由于时间、精力等多重缘故，管理端界面尚未制作，只能通过代码进行菜品的增添与管理人员的增加，实属一大遗憾。

在完成该系统的过程中，一直都遇到了不小的困难，但好在通过多方求助，终究是解决了。一是关于数据库的使用问题。因为是严格意义上第一次使用数据库，很多命令与 scheme 看不懂，开头便误删了几个 MySQL 自带的 scheme，导致后面许多命令都直接报错，只得将数据库卸载了之后再重新安装。随后，便是一直都无法使用 JDBC 连接数据库。经过层层筛选与排查，最后发现竟然是自己的接口名字出现了错误，白忙活了大半天。在连接上数据库之后，我便开始研读别人写的程序结构，力争能够把别人的代码框架读懂读透了。在研究完别人的框架之后，我便开始着手写自己的代码框架，这一步倒是蛮顺利的，毕竟有个参考能够依葫芦画瓢。随后，便是 UI 界面的设计。一开始的时候，对于各类排版与布局我是一窍不通的。在慢慢的模仿、尝试与学习之中，我也渐渐取得了一些进步。相较于我完成的第一个界面，我最后写的页面与其只差可谓是云泥之别。

通过这个程序我自己也真的学到了许多新的知识。可以说，也正是因为这次的机会，才能够让我更进一步的了解与熟悉 JAVA。

六 参考目录

[1] Xie, tansheng. "JavaSwing_1.3: GridBagLayout" CSDN. 05 May 2017.https://blog.csdn.net/xietansheng/article/details/72814552

[2] Baret-H. "MySQL 详细学习教程" CSDN 14 April 2021. https://bareth.blog.csdn.net/article/details/115712758?spm=1001.2014.3001.5506

[3] Xav, Zewen. "使用 JDBC 连接数据库" CSDN 08 Dec 2022https://blog.csdn.net/weixin_39591031/article/details/110900414?spm=1001.2014.3001.5506