|  |  |
| --- | --- |
| **实训编号** | **E11088** |
| **实训名称** | **Java语言课程实训** |
| **年级专业** | **2018级软件工程** |



**《Java语言课程实训》实训报告**

**客户资料管理系统的设计与实现**

学号及姓名： 18110506084 李威

18110506090 王飞

18110506105 王中豪

班 级： 软件1803班

完成日期： 2020年1月 8 日

**学生组内分工与贡献度**

|  |  |  |  |
| --- | --- | --- | --- |
| **姓名** | **班级** | **组内任务及分工**  **(按模块划分任务)** | **贡献度**  **（百分比）** |
| 李威 | 软件1803 | 主界面的所有功能和查询删除界面及代码 | 40 % |
| 王飞 | 软件1803 | 数据库构建，city，customer包的增加，修改，查询功能实现，和数据库的链接 | 30 % |
| 王中豪 | 软件1803 | 数据库构建，图标查找，添加界面及代码 | 30 % |
|  |  |  |  |

**成绩评定计分表**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 序号 | 学生姓名 | 需求分析设计方案 | 数据库  设计方案 | 文档  及排版 | 工作计划个人总结 | 代码  规范性 | 系统架构  合理性 | 系统实现系统测试 | 答辩  正确性 | 出勤情况  (缺勤1次减3分) | 合计 |
| 10分 | 10分 | 10分 | 10分 | 10分 | 10分 | 20分 | 20分 |
| 1 | 李威 |  |  |  |  |  |  |  |  |  |  |
| 2 | 王飞 |  |  |  |  |  |  |  |  |  |  |
| 3 | 王中豪 |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |

教师签名：

时间：2020年1月 日

**目 录**

目录

**[第一章 需求分析 1](#_Toc14372)**

[1.1 需求总体介绍 1](#_Toc31450)

[1.2 系统的功能性需求 1](#_Toc2216)

[1.3 系统的非功能性需求 1](#_Toc14236)

**[第二章 数据库设计 2](#_Toc21271)**

[2.1 数据库概念结构设计 2](#_Toc13136)

[2.2 数据库逻辑结构设计 4](#_Toc12811)

[2.3 数据库列表 5](#_Toc1795)

**[第三章 系统设计 6](#_Toc4914)**

[3.1 体系结构设计 6](#_Toc13585)

[3.2 人机交互设计 7](#_Toc1307)

[3.3 用户界面设计规则 8](#_Toc6456)

**[第四章 系统实现 9](#_Toc8358)**

[4.1 数据库实现 9](#_Toc3518)

[4.2 主界面的实现 10](#_Toc4153)

[4.3 增加功能的实现 13](#_Toc28591)

[4.4 查询功能的实现 14](#_Toc4108)

[4.5 修改功能的实现 15](#_Toc22634)

[4.6 删除功能的实现 16](#_Toc4571)

**[第五章 系统测试 17](#_Toc16664)**

[5.1 测试概要 17](#_Toc10020)

[5.2 测试结果及发现 17](#_Toc18023)

[5.3 测试结果 19](#_Toc11867)

**[第六章 系统使用说明书 19](#_Toc14197)**

[6.1 用户登录 19](#_Toc25275)

[6.2 进入系统 20](#_Toc12506)

[6.3 新增客户 21](#_Toc11164)

[6.3 访问客户列表修改 21](#_Toc19753)

# 需求分析

## 1.1 需求总体介绍

本项目为了方便维护客户信息为了方便维护客户信息，对客户进行统一管理，可以把所有客户信息录入系统，进行维护和统计功能。可通过文件的方式保存相关录入数据，对系统资料进行磁盘文件保存。旨在实现客户电子化管理，它主要包括客户界面、客户信息的增加界面、客户信息的修改界面、显示全部客户的信息、删除一条用户信息、按照用户的编号查询用户信息、按照用户的姓名查询用户信息、按照用户的地址查询用户信心、退出系统界面。客户管理系统是一个现代化的软件，它通过窗口swing和数据库的结合，以便于对客户信息集体化管理，提高了管理效率，美观性强。

## 1.2 系统的功能性需求

方便维护客户信息，对客户进行统一管理，把客户信息录入系统进行统计和管理维护。包括客户增加、客户删除、客户修改、客户查询、浏览客户、城市增加、城市删除、城市修改、城市查询、浏览城市、退出系统等功能。并且设置了两个用户权限等级，针对不同客户实现不同的功能。添加了天气，日期等功能并可以实时更新。

## 1.3 系统的非功能性需求

用户界面有增加用户，用户列表，修改密码等、需要配置好JAVA环境，对eclipse配置，连接好数据库，在数据库中建好列表。

# 第二章 数据库设计

## 2.1 数据库概念结构设计

等 级

性 别

等 级

城 市

客户信息

电 话

姓 名

编 号

图2-1 客户实体与其属性图

编 号

姓 名

电 话

增 加

城 市

等 级

性 别

等 级

图2-2增加客户信息图

## 

根据客户姓名

查 询

图2-3查询客户信息图

编 号

姓 名

电 话

城 市

修 改

等 级

等 级

性 别

图2-4修改客户信息图

输入客户编号删除

删 除

图2 -5删除客户信息图

客户管理系统

用户登录

查询

客户

信息

修改客户信息

删除客户信息

浏览

客户

信息

增加客户信息

退出系统

图2-6 总E-R图

## 2.2 数据库逻辑结构设计

#### 2.2.1完成关系模式的转化

基本信息：客户编号、客户姓名、客户电话、客户性别、客户邮箱、客户城市、客户等级

#### 2.2.2数据字典

数据和表的建立：

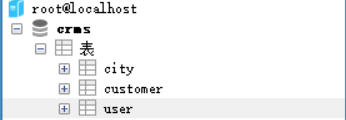


图2-7数据库建立图

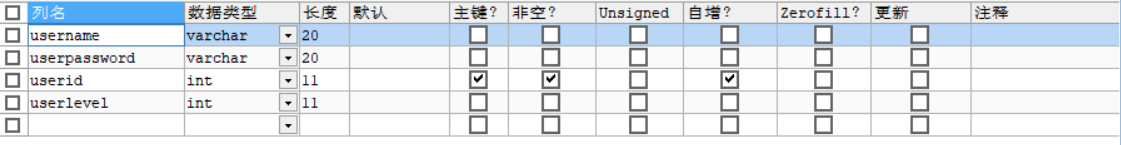


图2-8数据表建立图1

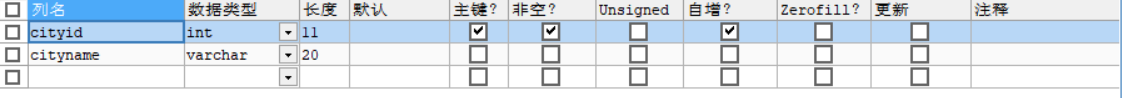


图2-9数据表建立图2

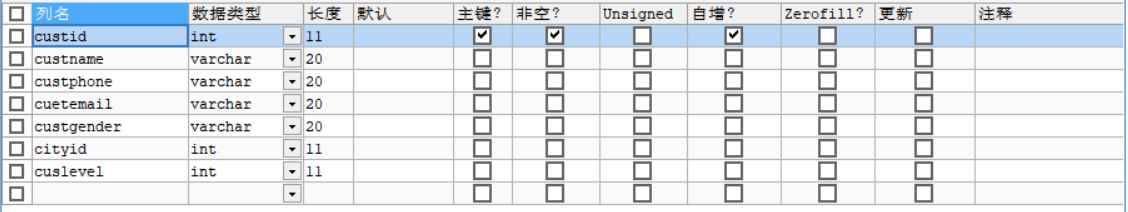


图2-10数据表建立图3

## 2.3 数据库列表

表2-1 城市信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段含义 |  |  |  |  |
| cityid | int | 城市编号 |  |  |  |  |
| cityname | varchar(20) | 城市姓名 |  |  |  |  |
|  |  |  |  |  |  |  |

表2-2 客户信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段含义 |  |  |  |  |
| custid | int | 客户编号 |  |  |  |  |
| custname | varchar | 客户姓名 |  |  |  |  |
| custphone | varchar | 客户电话 |  |  |  |  |
| custemail | varchar | 客户邮箱 |  |  |  |  |
| custgender | varchar | 客户性别 |  |  |  |  |
| cityid | int | 客户城市 |  |  |  |  |
| custlevel | int | 客户等级 |  |  |  |  |

图2-2 用户信息图

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 字段含义 |  |  |  |  |
| userid | int | 用户编号 |  |  |  |  |
| userpassword | varchar(20) | 用户密码 |  |  |  |  |
| username | varchar(20) | 用户姓名 |  |  |  |  |
| userlevel | int | 用户等级 |  |  |  |  |

# 第三章 系统设计

系统采用 dao设计。dao全称是data access object，数据库访问对象，主要的功能就是用于进行数据操作的，在程序的标准开发架构中属于数据层的操作。

## 3.1 体系结构设计

将需求分析中的系统功能用类（实体类、操作类、工具类、异常类等）的形式进行设计，搭建合理的系统架构。

dao包：

ICustomerDao类是用来装封一定功能（增、删、改、查、浏览）的集合。它好比一个模板，在其中定义了对象必须实现的成员，通过类或结构来实现它。

impl包：DAO接口的真实实现类，完成具体的数据库操作，但是不负责数据库的打开和关闭；

CustomerDao 类：用数据库来实现增删改查浏览等功能；

model包：

Customer类：客户属性get和set方法，toString方法

program包：程序界面

AddCustomerFrame类：增加客户的界面

CloseWindow 类：退出系统

DelCustomerFrame类：删除客户的界面

LoginFrame类：主界面，即登录上本系统弹出的界面

MainFrame类：主函数，运行管理系统

SearchCustomerFrame类：查询客户信息界面

ShowCustomerFrame类：显示所有员工界面

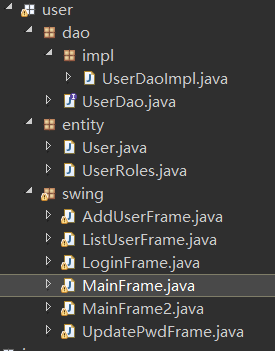
UpCustomerFrame类：修改客户信息的界面

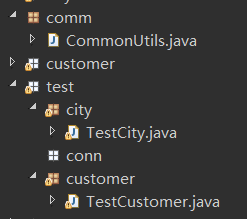
util包：

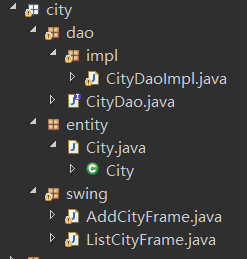
BaseDao类：数据库连接与关闭

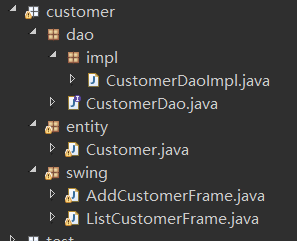
，

## 3.2 人机交互设计









## 3.3 用户界面设计规则

管理员管理部分：

1. 管理员可以增加客户信息
2. 管理员可以删除客户信息
3. 管理员可以修改客户信息
4. 管理员可以查询客户信息
5. 管理员可退出管理系统

# 第四章 系统实现

## 4.1 数据库实现

private String url = "jdbc:mysql://localhost:3306/cus?useUnicode=true&characterEncoding=UTF-8";

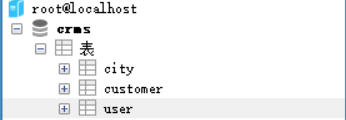
private String username = "root";

private String password = "";

private Connection conn = null;

建立一个数据库，数据库名为cus。

以下为用户登录，用户名如上语句为“root”，密码为null。



然后是数据库表的创建，表名为customer，如语句所示：

String sql = "SELECT \* FROM user WHERE username=? AND userpassword=?";

String sql = "UPDATE `user` SET `userpassword` = ? WHERE `user`.`userid` =?";

String sql = "SELECT \* FROM `user` WHERE username=?";

String sql = "UPDATE `user` SET `username` = ? WHERE `user`.`userid` =?";

String sql = "INSERT INTO `user` (`userid`, `username`, `userpassword`, `userlevel`) VALUES (NULL, ?, ?, ?)";

String sql = "SELECT \* FROM `user` WHERE userid=?";

String sql = "DELETE FROM `user` WHERE `user`.`userid` = ?";

String sql = "select \* from customer";

String sql = "SELECT \* FROM customer WHERE custid=?";

String sql = "INSERT INTO customer VALUES (NULL,?,?,?,?,?,?)";

String sql = "UPDATE customer SET custname = ?,custphone = ?,cuetemail = ?,custgender = ?,cityid = ?,cuslevel = ? WHERE custid = ?";

String sql = "DELETE FROM customer WHERE custid=?";

String sql = "select \* from city";

String sql = "select \* from city order by cityid desc";

String sql = "SELECT \* FROM city WHERE cityid = ?";

String sql = "INSERT INTO city VALUES (NULL,?)";

String sql = "UPDATE city SET cityname = ? WHERE cityid = ?";

String sql = "DELETE FROM city WHERE cityid=?";

String sql = "SELECT \* FROM city WHERE cityname = ?";

## 4.2 主界面的实现

protected void listCustomerButton(MouseEvent e) {

new ListCustomerFrame().setVisible(true);

}

/\*\*

\* 新增客户

\* **@param** e

\*/

protected void addCustomerButton(MouseEvent e) {

new AddCustomerFrame().setVisible(true);

}

/\*\*

\* 新增城市

\* **@param** e

\*/

protected void addCityButton(MouseEvent e) {

new AddCityFrame().setVisible(true);

}

/\*\*

\* 城市列表

\* **@param** e

\*/

protected void listCityButton(MouseEvent e) {

new ListCityFrame().setVisible(true);

}

/\*\*

\* 用户列表按钮点击事件

\* **@param** e

\*/

protected void listUserButton(MouseEvent e) {

new ListUserFrame().setVisible(true);

}

/\*\*

\* 增加用户按钮点击事件

\* **@param** e

\*/

protected void addUserButton(MouseEvent e) {

new AddUserFrame().setVisible(true);

}

/\*\*

\* 客户管理组件点击事件

\* **@param** e

\*/

protected void mangeCusButton(MouseEvent e, JPanel mangeCusPanel) {

getContentPane().add(mangeCusPanel);

}

/\*\*

\* 退出按钮

\* **@param** e

\*/

protected void exitButton(ActionEvent e) {

System.*exit*(0);

}

/\*\*

\* 修改密码按钮

\* **@param** arg0

\* **@param** user

\*/

protected void updatePasswordButton(ActionEvent arg0, User user) {

new UpdatePwdFrame(user).setVisible(true);

}

/\*\*

\* 窗口大小改变事件

\*/

protected void mainFrameComponentResized(ComponentEvent arg0, JPanel topPanel, GroupLayout gl\_panel) {

}

## 4.3 增加功能的实现

protected void addButton(ActionEvent e) {

String cityname = this.citynameText.getText();}

if(cityname.isEmpty()) {

JOptionPane.*showMessageDialog*(this, "请输入城市名称");

this.citynameText.requestFocus();

return ;

}

if(cityname.length() > 20) {

JOptionPane.*showMessageDialog*(this, "对不起，城市名称不能超过20位！");

this.citynameText.requestFocus();

return ;

}

City city = new City();

city.setCityname(cityname);

int row = cdao.addCity(city);

if(row > 0) {

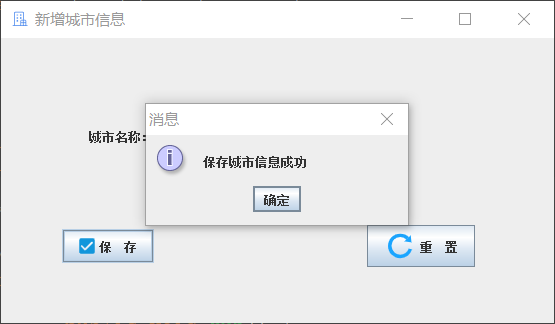
JOptionPane.*showMessageDialog*(this, "保存城市信息成功");

}else {

JOptionPane.*showMessageDialog*(this, "保存城市信息失败");

}

}



## 4.4 查询功能的实现

protected void detailCustomer(MouseEvent arg0) {

int row = this.table.getSelectedRow();

int custid = (Integer)this.table.getValueAt(row, 0);

Customer customer = cdao.queryCustomerById(custid);

this.custidText.setText(String.*valueOf*(customer.getCustid()));

this.custnameText.setText(customer.getCustname());

this.custphoneText.setText(customer.getCustphone());

this.cuetemailText.setText(customer.getCuetemail());

this.getCitys("modify");

this.cuslevelText.setText(String.*valueOf*(customer.getCuslevel()));

if(customer.getCustgender().equals("男")) {

this.menjrbn.setSelected(true);

}else {

this.womenjrb.setSelected(true);

}

int count = this.cityidJrb.getItemCount();

for(int i = 0;i < count;i++) {

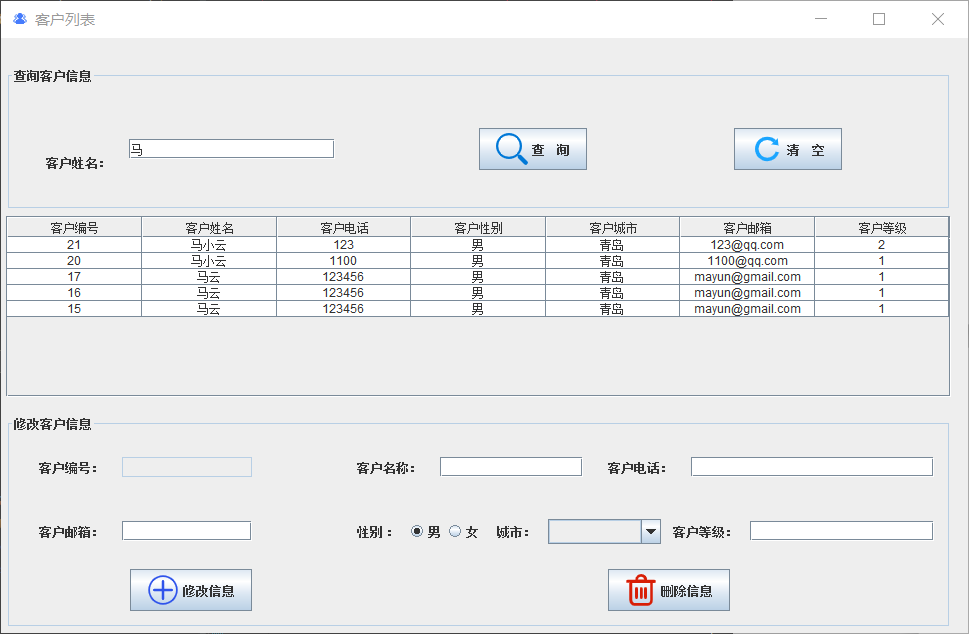
City selectcity = dao.queryCityById(customer.getCityid());

if(selectcity.getCityname().equals(this.cityidJrb.getItemAt(i))) {

this.cityidJrb.setSelectedIndex(i);

}

}

}

## 4.5 修改功能的实现

protected void gainameButton(ActionEvent e) {

String strCustid = this.custidText.getText();

int custid = Integer.*parseInt*(strCustid);

String custname = this.custnameText.getText();

String cityname = (String) this.cityidJrb.getItemAt(cityidJrb.getSelectedIndex());

String cuetemail = this.cuetemailText.getText();

String custphone = this.custphoneText.getText();

City city = dao.queryCityByName(cityname);

int cuslevel = Integer.*parseInt*(this.cuslevelText.getText());

Customer customer = new Customer();

customer.setCustid(custid);

customer.setCustname(custname);

customer.setCuetemail(cuetemail);

customer.setCityid(city.getCityid());

customer.setCuslevel(cuslevel);

customer.setCustphone(custphone);

if(this.menjrbn.isSelected()) {

customer.setCustgender("男");

}else {

customer.setCustgender("女");

}

int row = cdao.updateCustomer(customer);

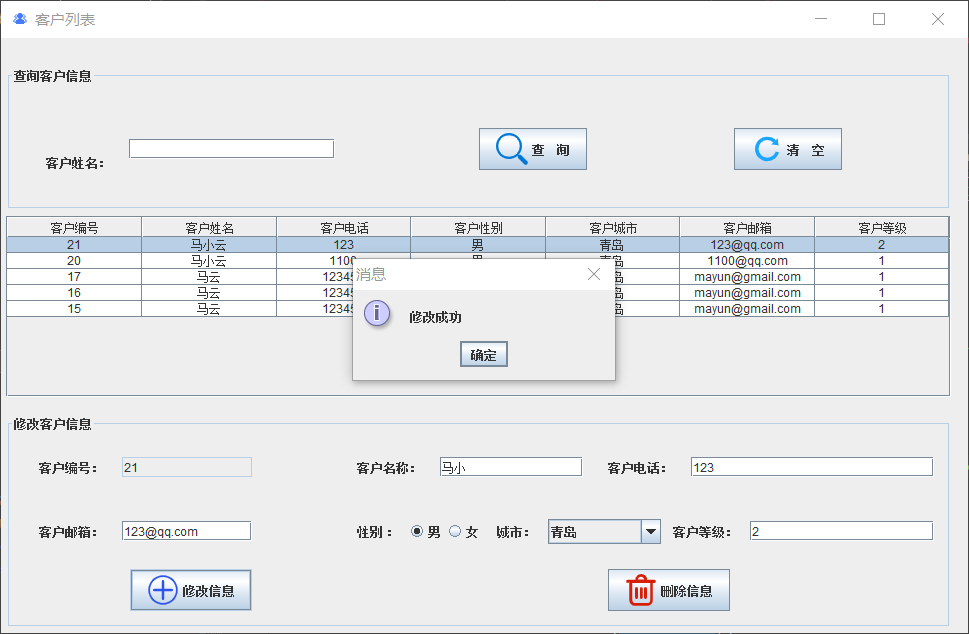
if(row > 0) {

JOptionPane.*showMessageDialog*(this, "修改成功");

this.getCustomers(null);

}else {

JOptionPane.*showMessageDialog*(this, "修改失败");

}

}

## 4.6 删除功能的实现

protected void deleButton(ActionEvent e) {

String strCustid = this.custidText.getText();

int custid = Integer.*parseInt*(strCustid);

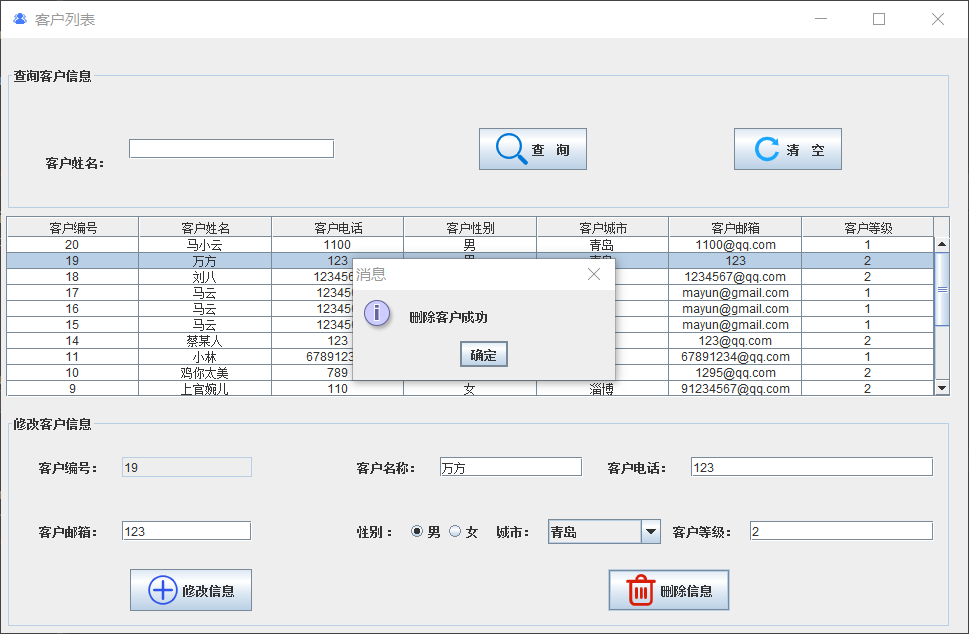
int row = this.cdao.deleteCustomer(custid);

if(row > 0) {

JOptionPane.*showMessageDialog*(this, "删除客户成功");

this.getCustomers(null);

}else {

JOptionPane.*showMessageDialog*(this, "删除客户失败");

}

}

# 第五章 系统测试

## 5.1 测试概要

先建立数据库测试，然后分别对客户增加、客户删除、客户修改、客户查询、浏览客户、等功能进行测试。

## 5.2 测试结果及发现

表5-1建立数据库测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 1 | | |
| 控件名称 |  | | |
| 描述 | 建立数据库 | | |
| **编号** | **输入/动作** | **期望的输出** | **实际情况** |
| 1 | 输入用户名“root”，密码为null | 无输出 | 提示“数据库建立成功” |

表5-2增加客户测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 2 | | |
| 控件名称 |  | | |
| 描述 | 增加客户 | | |
| **编号** | **输入/动作** | **期望的输出** | **实际情况** |
| 1 | 编号，姓名 | 提示“添加成功！” | 提示“添加成功！” |
| 2 | 编号，地址 | 提示“添加失败！” | 提示“添加失败！” |
| 3 | 姓名，地址 | 提示“添加失败！” | 提示“添加失败！” |
| 4 | 编号，地址，姓名 | 提示“添加成功！” | 提示“添加成功！” |

表5-3删除客户测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 3 | | |
| 控件名称 |  | | |
| 描述 | 删除客户 | | |
| **编号** | **输入/动作** | **期望的输出** | **实际情况** |
| 1 | 不存在的编号 | 提示“查询无此客户！” | 提示“查询无此客户！” |
| 2 | 存在的编号 | 提示“删除成功！” | 提示“删除成功！” |

表5-4修改客户测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 4 | | |
| 控件名称 |  | | |
| 描述 | 修改客户 | | |
| **编号** | **输入/动作** | **期望的输出** | **实际情况** |
| 1 | 不存在的编号 | 提示“查询无此客户！” | 提示“查询无此客户！” |
| 2 | 存在的编号，输入姓名和地址 | 提示“修改成功！” | 提示“修改成功！” |

表5-5 查询客户测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 5 | | |
| 控件名称 |  | | |
| 描述 | 查询客户 | | |
| **编号** | **输入/动作** | **期望的输出** | **实际情况** |
| 1 | 编号 | 客户姓名和地址 | 客户姓名和地址 |
| 2 | 地址 | 客户编号和姓名 | 客户编号和姓名 |
| 3 | 姓名 | 客户编号和地址 | 客户编号和地址 |
| 4 | 不存在的编号，姓名，或者地址 | 提示“查询无此客户！” | 提示“查询无此客户！” |

表5-6浏览客户测试表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 6 | | |
| 控件名称 |  | | |
| 描述 | 浏览客户 | | |
| **编号** | **输入/动作** | **期望的输出** | **实际情况** |
| 1 | 无输入 | 所有客户信息，包括无客户的情况 | 所有客户信息，包括无客户的情况 |

## 5.3 测试结果

经测试，数据库，客户增加、客户删除、客户修改、客户查询、浏览客户、等功能

都正常，可正常运行、

# 第六章 系统使用说明书

## 6.1 用户登录

首先输入用户名密码登陆系统

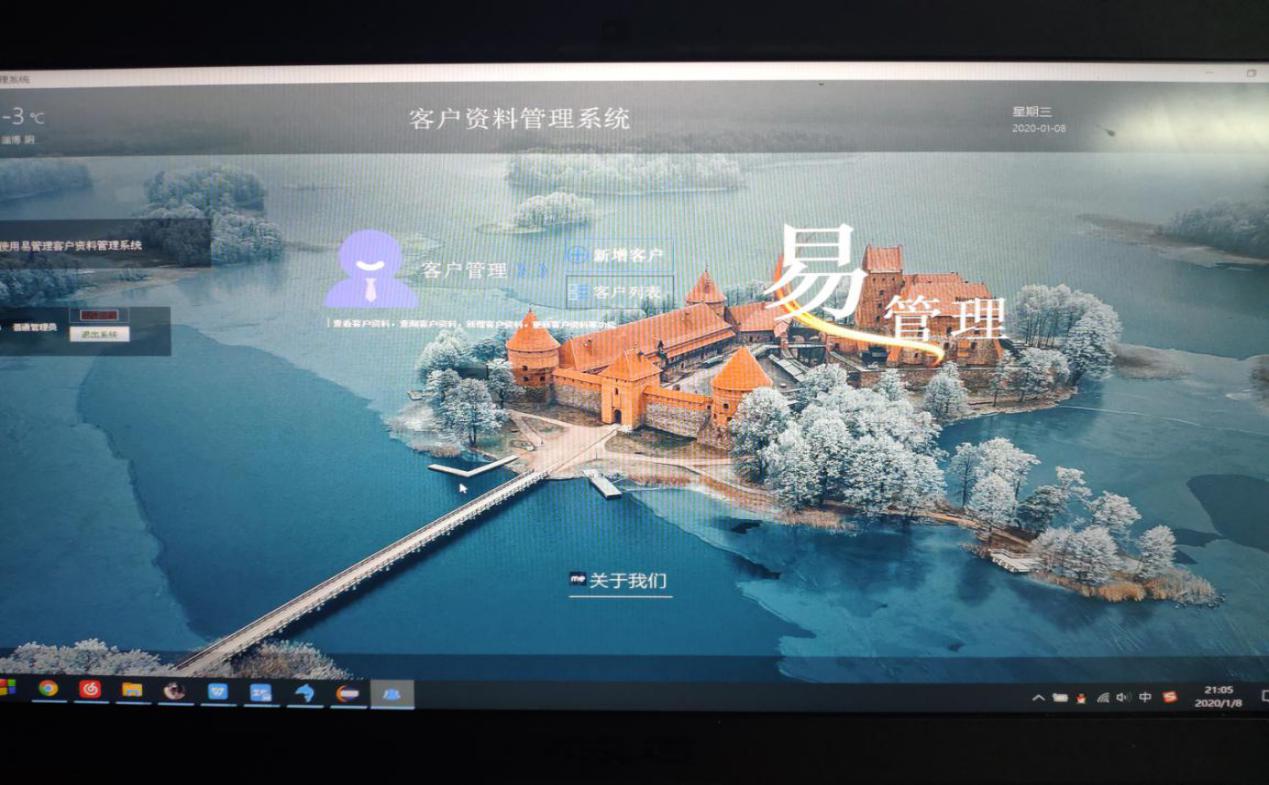


## 

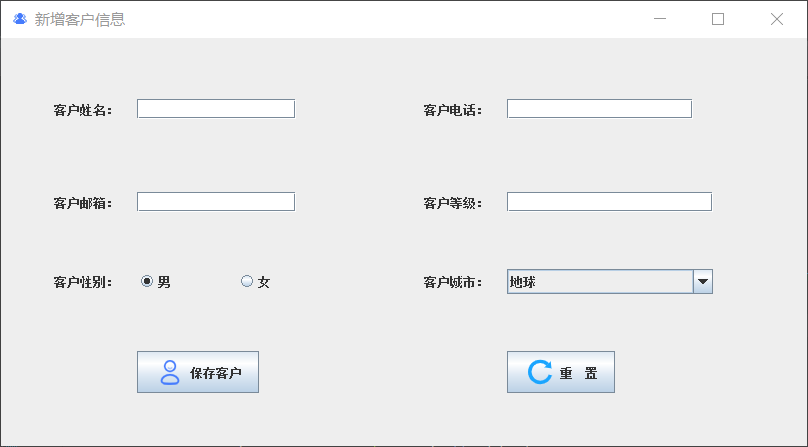
## 6.2 进入系统

两种权限对应两种登录界面，实现不同的功能





## 6.3 新增客户



## 6.3 访问客户列表修改

