**福州理工学院**

课程期末（报告）

|  |  |
| --- | --- |
| 题 目： | Boot客户管理系统 |
| 学 生： |  |
| 指导老师： | 吴献 |
| 院 别： |  |
| 专 业： |  |
| 班 级： |  |
| 学 号： |  |

2023年12月

目录

[一、 课程设计的目的 1](#_Toc4599)

[二、 课程设计题目及要求 1](#_Toc7418)

[2.1 课程设计题目 1](#_Toc12189)

[2.2 课程设计要求 1](#_Toc18008)

[三、 程序设计思路 1](#_Toc6444)

[3.1选题和计划 1](#_Toc4822)

[3.1.1选题 1](#_Toc27099)

[3.1.2计划 2](#_Toc2415)

[3.2需求分析 3](#_Toc10939)

[3.2.1需求 3](#_Toc15599)

[四、 系统设计 8](#_Toc1026)

[4.1新增客户 8](#_Toc13331)

[4.2查询客户 8](#_Toc2069)

[4.3注册会员 10](#_Toc9102)

[4.4会员信息 11](#_Toc12377)

[4.5客户反馈 11](#_Toc833)

[4.6历史记录 11](#_Toc20802)

[4.7退出系统 11](#_Toc29472)

[五、 项目操作手册 12](#_Toc1175)

[六、 小结 33](#_Toc30651)



## 一、课程设计的目的

课程设计围绕课程的整体教学目标进行，具有一定的综合性和应用性，只靠个别单元的知识一般不能实现课程设计的目标，需要对课程知识进行综合运用。要完成好课程设计，就要对课程知识进行系统学习、复习、总结、应用，达到在实践中学习知识、掌握知识、应用知识和培养能力的目的。本次课程设计旨在创建一款高度专业、全面的Boot客户管理系统，以解决企业在客户关系管理方面的挑战。通过该系统的开发，旨在实现以下长期目标：

1. 提升客户服务质量：

通过系统化管理客户信息和跟进记录，企业能够更迅速、准确地响应客户需求，从而提高客户服务质量。系统将提供一站式的客户信息汇总，有助于团队更全面地理解客户背景，以更个性化、精准的方式满足客户期望。

2. 提高销售效率：

通过实现销售机会的精细化管理和销售漏斗的可视化，系统旨在帮助销售团队更好地跟踪潜在业务，准确预测销售机会，最终提高销售效率。可视化的销售漏斗将使销售团队更容易识别和处理潜在的问题，从而更灵活地应对市场变化。

3. 优化客户关系：

通过系统的分类和标签管理，企业可以更细致地了解不同类型客户的需求和喜好，为不同客户提供个性化服务。跟进记录的完善将有助于建立更紧密的客户关系，提高客户忠诚度，最终实现客户满意度的全面提升。

## 二、课程设计题目及要求

2.1 课程设计题目

课程设计的题目为《Boot客户管理系统》，设计一个全面的Boot客户管理系统，以满足企业对客户信息的有效管理和跟踪需求，提高客户服务质量和效率。以解决企业在客户关系管理方面的挑战。

2.2 课程设计要求

1. 独立完成项目及报告，**用户管理模块：**
   * 实现用户的注册、登录、修改密码等基本功能。
   * 区分不同用户权限，如管理员和普通用户。
2. **客户信息管理模块：**
   * 实现客户基本信息的录入、修改、删除功能。
   * 支持客户分类和标签管理，方便快速检索和分析客户群体。
3. **跟进记录模块：**
   * 记录客户的沟通历史，包括拜访、电话、邮件等形式。
   * 提供跟进提醒功能，确保关键时刻不漏掉任何客户。
4. **销售机会管理模块：**
   * 追踪潜在销售机会，记录关键信息，如预计成交时间、预计销售额等。
   * 支持销售漏斗的可视化，方便销售团队分析和决策。
5. **报表统计模块：**
   * 提供客户数量、销售额、销售漏斗等关键指标的报表统计功能。
   * 支持导出报表数据，方便进一步分析和决策。

## 三、程序设计思路

3.1选题和计划

3.1.1选题

选取了《Boot客户管理系统》作为课程设计的题目。在当今竞争激烈的商业环境中，客户关系管理对企业的成功至关重要。随着市场的不断变化和客户需求的日益多样化，传统的客户管理方式已经无法满足企业的需求。因此，本课程设计选题《Boot客户管理系统》的提出具有重要的背景和深远的意义。

3.1.2计划

2023-12-15~2023-12-21完成

3.2需求分析

3.2.1需求

实现客户管理系统的基本功能。比如新增客户，客户信息查询，会员管理，日志查询等操作。详细介绍如下：

新增客户信息：姓名，电话 ，住址，会员等级

客户信息查询：按照姓名进行查找

会员管理：查询会员的信息

日志操作：查看每一步操作的时间及干了啥。

## 四、系统设计

4.1系统分层

1.客户信息层

客户信息，反馈信息

2.会员层

会员信息，会员积分

4.2管理模块分析

系统设计包括新增客户、查询客户、注册会员、会员信息、客户反馈、历史记录、退出系统等模块。下面给出了客户类的实现和一部分系统主函数的代码。

杂项头文件

#pragma once

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include<stdio.h>

#include<sstream>

#include<time.h>

#include<stdlib.h>

#include <chrono>

using namespace std;

struct record\_t

{

record\_t(string name,string did,string time) : \_name(name), \_did(did), time(time) {}

string \_name;//客户名字

string \_did;//干了啥事

string time;//时间

};

struct search

{

search(string name, string phone, string address, int vip, int integral, string ret)

:\_name(name)

, \_phone(phone)

, \_address(address)

, \_vip(vip)

, \_integral(integral)

, \_retVal(ret){}

string \_name;//客户名字

string \_phone;//客户手机号

string \_address;//客户地址

int \_vip;//会员等级

int \_integral;//积分

string \_retVal;//客户反馈

};

客户类的定义

#ifndef CUSTOMER\_H

#define CUSTOMER\_H

#include"head.h"

//客户类

class Customer {

public:

//构造函数

Customer(string name, string phone, string address, int vip = 0, int integral = 0,string ret="");

//客户信息

void getInfo();

//会员注册

void SignUpVip(int vip);

//客户反馈

void returnVal(string ret);

string getName();

int getVip();

void setVip(int vip);

string OfsFile();

~Customer();

private:

string \_name;//客户名字

string \_phone;//客户手机号

string \_address;//客户地址

int \_vip;//会员等级

int \_integral;//积分

string \_retVal;//客户反馈

};

#endif // CUSTOMER\_H

客户类的实现

#include "Customer.h"

Customer::Customer(string name, string phone, string address, int vip , int integral ,string ret)

:\_name(name)

, \_phone(phone)

, \_address(address)

, \_vip(vip)

, \_integral(integral)

,\_retVal(ret)

{

ofstream ofs("customer.txt", std::ios::app);

if (!ofs)

{

cerr << "open fail" << endl;

return;

}

ofs << \_name << "\t" << \_phone << "\t" << \_address << "\t" << \_vip << "\t" << \_integral << "\t"

<< ret << endl;

ofs.close();

}

Customer::~Customer()

{

}

//客户信息

void Customer::getInfo()

{

cout <<"姓名:" << \_name <<"\t电话:" << \_phone <<"\t地址:" << \_address<<"\tvip等级:"

<< \_vip<<"\t会员积分:" << \_integral << endl;

}

//会员注册

void Customer::SignUpVip(int vip)

{

\_vip = vip;

}

//客户反馈

void Customer::returnVal(string retVal)

{

\_retVal = retVal;

}

string Customer::getName()

{

return \_name;

}

int Customer::getVip()

{

return \_vip;

}

void Customer::setVip(int vip)

{

\_vip = vip;

}

string Customer::OfsFile()

{

// 使用 std::to\_string() 将整型转换为字符串

std::string s = \_name + "\t" + \_phone + "\t" + \_address + "\t" +

std::to\_string(\_vip) + "\t" + std::to\_string(\_integral);

return s;

}

主函数

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include "Customer.h"

//定义全局变量

vector<record\_t>records;

vector<Customer>cust;

vector<search>Sinfo;

void ofsRecod()

{

ofstream ofs("record.txt", std::ios::app);

if (!ofs)

{

cerr << "open file" << endl;

return;

}

for (auto& x : records)

{

ofs << x.\_name << "\t" << x.\_did << "\t" << x.time << endl;

}

ofs.close();

}

void readRecord()

{

ifstream ifs("record.txt");

if (!ifs)

{

cerr << "open file" << endl;

return;

}

cout << "姓名\t干了啥事\t时间" << endl;

string line;

while (getline(ifs, line))

{

cout << line << endl;;

}

ifs.close();

}

string getCurrentTime()

{

// 获取当前时间点

auto now = std::chrono::system\_clock::now();

// 将当前时间点转换为time\_t类型

std::time\_t currentTime = std::chrono::system\_clock::to\_time\_t(now);

// 使用std::localtime函数将time\_t类型的时间转换为tm结构体，便于提取年月日时分秒

std::tm\* localTime = std::localtime(&currentTime);

// 格式化输出

char buffer[20];

std::strftime(buffer, sizeof(buffer), "%Y-%m-%d-%H-%M", localTime);

return buffer;

}

void menu()

{

printf(" Boot客户管理系统 \n");

printf(" 1.新增客户 \n");

printf(" 2.查询客户 \n");

printf(" 3.注册会员 \n");

printf(" 4.会员信息 \n");

printf(" 5.客户反馈 \n");

printf(" 6.历史记录 \n");

printf(" 0.退出系统 \n");

}

int cinInfo()

{

int num = 0;

while (cin >> num, !cin.eof())

{

if (cin.bad())

{

cout << "cin has breken!" << endl;

return -1;

}

else if (cin.fail())

{

cin.clear();

cin.ignore(std::numeric\_limits<std::streamsize>::max(), '\n');

cout << "pls input your choice" << endl;

}

else

{

break;

}

}

return num;

}

void newPeo()

{

string name, phone, address, ret;

int vip = 0, integral = 0;

cout << "请输入姓名 #";

cin >> name;

cout << "请输入电话 #";

cin >> phone;

cout << "请输入地址 #";

cin >> address;

cout << "是否注册会员输入VIP等级即可(0为不注册) #";

cin >> vip;

integral \*= vip;

cust.push\_back(Customer(name, phone, address, vip, integral));

if(vip==0)

records.push\_back(record\_t(name,"填写信息未注册会员", getCurrentTime()));

else

records.push\_back(record\_t(name, "填写信息并注册会员"+std::to\_string(vip)+"级", getCurrentTime()));

ofsRecod();

records.clear();

cust.clear();

cout << "添加成功" << endl;

}

void searchInfo()

{

string name, phone, address, ret;

string vip , integral;

if (cust.empty())

{

//就说明再次打开系统需要加载信息

cout << "加载信息..." << endl;

ifstream ifs("customer.txt");

if (!ifs)

{

cerr << "open fail" << endl;

return;

}

string line;

while (getline(ifs, line))

{

istringstream iss(line);

iss >> name >> phone >> address >> vip >> integral >> ret;

Sinfo.push\_back(search(name, phone, address, atoi(vip.c\_str()), atoi(integral.c\_str()), ret));

}

ifs.close();

}

cout << "加载成功" << endl;

cout << "请输入客户姓名 #";

cin >> name;

for (auto& x : Sinfo)

{

if (x.\_name == name)

{

cout << "姓名:" << x.\_name << "\t电话:" << x.\_phone << "\t地址:" << x.\_address << "\tvip等级:"

<< x.\_vip << "\t会员积分:" << x.\_integral << endl;

records.push\_back(record\_t(name, "查询信息并且找到了", getCurrentTime()));

ofsRecod();

records.clear();

Sinfo.clear();

return;

}

}

cout << "未找到" << endl;

records.push\_back(record\_t(name, "查询信息并且未找到", getCurrentTime()));

ofsRecod();

records.clear();

Sinfo.clear();

}

void ofsVip()

{

ofstream ofs("customer.txt");

if (!ofs)

{

cerr << "open fail" << endl;

return;

}

for (auto& x : Sinfo)

{

ofs << x.\_name << "\t" << x.\_phone << "\t" << x.\_address << "\t" << x.\_vip << "\t" << x.\_integral << "\t"

<< x.\_retVal << endl;

}

ofs.close();

}

void signVip()

{

string name, phone, address, ret;

int vip = 0, integral = 0;

if (cust.empty())

{

//就说明再次打开系统需要加载信息

ifstream ifs("customer.txt");

if (!ifs)

{

cerr << "open fail" << endl;

return;

}

string line;

while (getline(ifs, line))

{

istringstream iss(line);

iss >> name >> phone >> address >> vip >> integral >> ret;

Sinfo.push\_back(search(name, phone, address, vip, integral, ret));

}

ifs.close();

}

cout << "请输入客户姓名 #";

cin >> name;

for (auto& x : Sinfo)

{

if (x.\_name==name && x.\_vip==0)

{

cout << "注册会员几级 #";

cin >> vip;

x.\_vip=vip;

break;

}

else if (x.\_name == name && x.\_vip != 0)

{

cout << "你已经是vip用户且当前是vip" << x.\_vip << "级" << endl;

cout << "是否升级,输入1或0（不需要）" << endl;

int cmd = -1;

cin >> cmd;

if (cmd == 0)

{

break;

}

cout << "升到几级 #";

cin >> vip;

x.\_vip = vip;

break;

}

}

//重新加载到文件里

ofsVip();

Sinfo.clear();

}

void vipInfo()

{

string name, phone, address, ret;

int vip = 0, integral = 0;

if (cust.empty())

{

//就说明再次打开系统需要加载信息

ifstream ifs("customer.txt");

if (!ifs)

{

cerr << "open fail" << endl;

return;

}

string line;

while (getline(ifs, line))

{

istringstream iss(line);

iss >> name >> phone >> address >> vip >> integral >> ret;

Sinfo.push\_back(search(name, phone, address, vip, integral, ret));

}

ifs.close();

}

for (auto& x : Sinfo)

{

if (x.\_vip != 0)

{

cout << "姓名:" << x.\_name << "\t电话:" << x.\_phone << "\t地址:" << x.\_address << "\tvip等级:"

<< x.\_vip << "\t会员积分:" << x.\_integral << endl;

}

}

Sinfo.clear();

}

void logRet()

{

string name, phone, address, ret;

int vip = 0, integral = 0;

if (cust.empty())

{

//就说明再次打开系统需要加载信息

ifstream ifs("customer.txt");

if (!ifs)

{

cerr << "open fail" << endl;

return;

}

string line;

while (getline(ifs, line))

{

istringstream iss(line);

iss >> name >> phone >> address >> vip >> integral >> ret;

Sinfo.push\_back(search(name, phone, address, vip, integral, ret));

}

ifs.close();

}

cout << "输入你的姓名 #";

cin >> name;

cout << "你要反馈的信息 #";

cin >> ret;

for (auto& x : Sinfo)

{

if (x.\_name== name)

{

x.\_retVal=ret;

break;

}

}

//重新保存信息

ofstream ofs("customer.txt");

if (!ofs)

{

cerr << "open fail" << endl;

return;

}

for (auto& x : Sinfo)

{

ofs << x.\_name << "\t" << x.\_phone << "\t" << x.\_address << "\t" << x.\_vip << "\t" << x.\_integral << "\t"

<< x.\_retVal << endl;

}

ofs.close();

Sinfo.clear();

}

int main()

{

while (1)

{

menu();

cout << "请输入你的命令(0~6) #";

int num = cinInfo();

switch (num)

{

case 1:

newPeo();

break;

case 2:

searchInfo();

break;

case 3:

signVip();

break;

case 4:

vipInfo();

break;

case 5:

logRet();

break;

case 6:

readRecord();

break;

case 0:

exit(0);

default:

cout << "命令不存在" << endl;

continue;

}

system("pause");

system("cls");

}

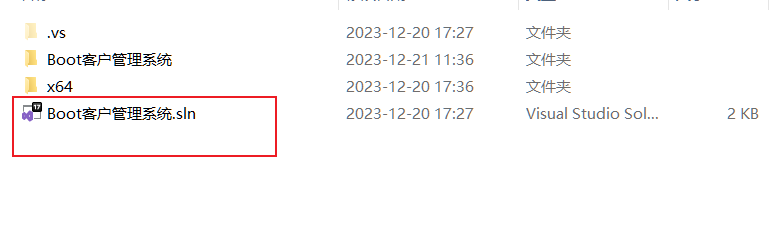
return 0;

}

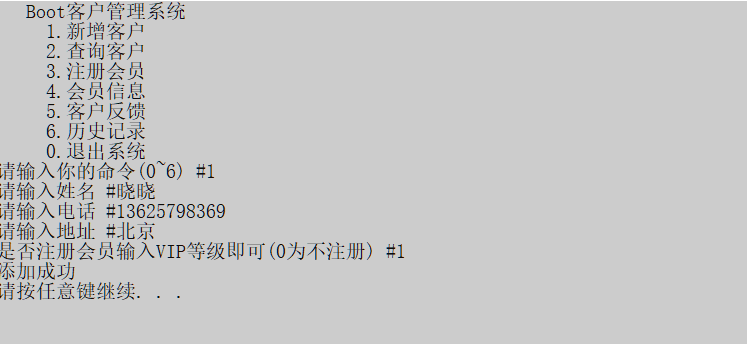
## 五、项目操作手册

下载vs2022

点击



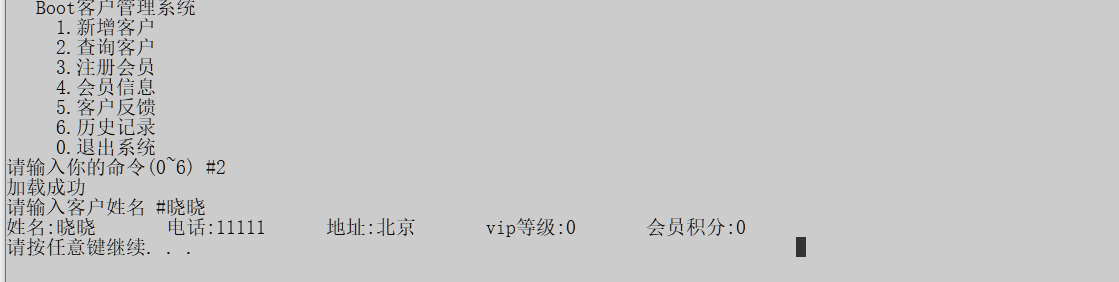
直接就可以跳转到vs2022中进行相关操作

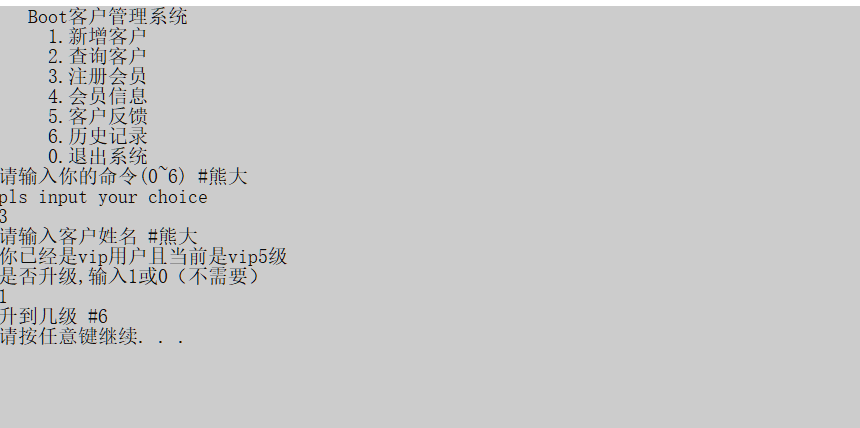


图表, 散点图

描述已自动生成

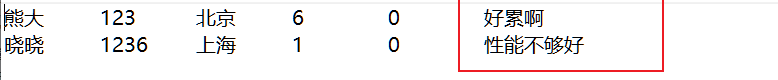
Bug已改好





图片包含 形状

描述已自动生成



散点图

低可信度描述已自动生成

## 六、小结

计算机技术的飞速发展带动了很多行业的发展，计算机带给我们最直观的价值就是减少了人力物力的投资，提高了工作质量和效率，从而提高了整个社会的生产力。

在本项目的运行中也发现代码的编译过程中失之毫厘差之千里，一定要注意好代码的完整性，数据的严谨性，做到有头有尾，完完整整。

最后我想说的是，写代码做开发一定要有耐心，要仔细检查代码中的问题，反复进行调试，确保实现功能完善最优化。感谢吴献老师不遗余力，呕心沥血的指导！