**《机票预订系统退票模块测试分析报告》**

1、编写目的

测试分析报告是在测试分析的基础上，对测试结果以及测试的数据等加以记录和分析总结。它也是测试过程中的一个重要环节。同时，它也是对软件性能的一个总的分析和认可及对不足之处的说明。因此，测试分析报告对于今后对软件的功能的加强，不足之处的弥补等都起着十分重要的提纲作用。另外，它还有利于今后软件开发者阅读原程序，根据测试提供的数据和结果，分析原代码，掌握各函数的功能和局限性。从而缩短软件开发者的再开发时间和所耗费的精力、资金。从这方面上，我们不难发现，测试分析报告所指明的对象主要是针对于软件开发者。

2.测试源码：

void search\_delete()

{

node \*p, \*q, \*r;

wait \*tempw, \*tempw2, \*tempw3;

int step = 1, t1, t2, i;

char ch;

p = new node;

tempw = new wait;

tempw2 = new wait;

tempw3 = new wait;

q = head1;

cout << endl << "请输入个人信息" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

cout << endl << "姓名 : "; cin >> p->name;

do{

q = q->next;

if ((q != NULL) &&

(comp(q, p)))

{

cout << endl;

q->date = q->plane;

cout << "Located!" << endl;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout << endl << "姓名 : " << q->name;

cout << endl << "ID号 : " << q->id;

cout << endl << "座位号 : " << q->seat;

cout << endl << "班机号 : " << q->plane;

cout << endl << "日期 : " << q->date << endl;

cout << "删除该纪录 ? [Y/N] ";

cin >> ch;

if (ch == 'Y' || ch == 'y') {

t1 = q->plane;

t2 = q->seat;

a[t1].seat[t2] = 0;

a[t1].seat[0]--;

r = q; q = q->pre;

r->pre->next = r->next;

if (r->next != NULL) r->next->pre = r->pre;

delete(r);

cout << "\*\*\*\* 记录删除成功 ! \*\*\*\*";

write\_to\_file();

tempw = wait\_head;

for (i = 0; i<wait\_head->count; i++)

{

tempw = tempw->next;

if (tempw == NULL) break;

if ((tempw->plane == t1) && (tempw->seat == t2))

{

strcpy(tempw3->name, tempw->name);

strcpy(tempw3->phone, tempw->phone);

cout << endl << "等候的人中有可以订票的了：" << endl;

cout << endl << "姓名 : " << tempw->name;

cout << endl << "ID号 : " << tempw->id << endl;

a[t1].seat[0]++;

a[t1].seat[t2] = 1;

if (tempw->plane == 1) add\_information(head1, 1, tempw->seat);

if (tempw->plane == 2) add\_information(head2, 2, tempw->seat);

if (tempw->plane == 3) add\_information(head3, 3, tempw->seat);

tempw2 = tempw->pre;

tempw2->next = tempw->next;

if (tempw->next == NULL) wait\_end = tempw2;

else tempw->next->pre = tempw2;

delete(tempw);

wait\_head->count--;

write\_to\_file();

cout << endl << "等候的" << tempw3->name << "已经成功订票，已经由电话" << tempw3->phone << "通知了" << endl;

break;

}

}

}

}

} while (1);

}

3.测试项目

3、 缺陷与限制

网络传输的保密性，安全性以及数据库的安全性还存在一定的缺陷，有可能会由于传输过程中数据的丢失造成软件运行错误。

4、经验和体会

1、测试用例要根据测试计划来编写。

　　2、测试用例也要分测试项进行归类，这样比较好分析和阅读。

　　3、编写测试用例要考虑各种情况，精力主要集中在软件的主要业务流程和风险高的地方。能分出测试优先级别就最好了。

　　4、熟悉系统，对编写测试用例很有帮助。