**实验二：小型机票订票系统设计**

**学号：22920142203809 姓名：冯立刚 年级：2014级**

[**一、** **实验目的：** 2](#_Toc466495303)

[**二、** **实验内容：** 2](#_Toc466495304)

[**三、** **实验步骤：** 2](#_Toc466495305)

[**1、** **建立关系模型，用ER图表示如下：** 3](#_Toc466495306)

[**2、** **定义数据库表** 3](#_Toc466495307)

[**3、** **软件功能实现** 5](#_Toc466495308)

[**4、** **相关实现代码说明：** 6](#_Toc466495309)

[**四、 实验系统演示： 11**](#_Toc466495310)

[**五、 实验总结： 18**](#_Toc466495311)

1. **实验目的：**

1、能够正确运用数据库基本理论和知识，复习、巩固、提高数据库方案设计、论证和分析方法。

2、熟悉关系数据库规范化设计理论，根据实验要求设计并建立科学合理的数据库，正确建立数据库中表与表之间的关系。

3、进一步正确理解数据库设计思路，培养分析问题、解决问题的能力，提高查询资料和撰写书面文件的能力。

1. **实验内容：**

[系统描述]

小型机票订票系统应具备航班信息管理、旅客信息管理、机票信息查询及订购基本管理功能，具体要求如下：

（1）能对航班信息进行维护：输入、修改、删除航班信息。

（2）能对旅客信息进行维护。

（3）能进行航班信息、余票情况及旅客信息的各种查询。

（4）能进行机票的订购及退订，必须动态刷新库存。

（5）能对每个航班进行按月、季度、年份进行售票统计。

（6）能对每个旅客进行按月、季度、年份进行出行统计。

1．根据上述系统功能需求，使用ER图描述该管理信息系统的概念模型。

2．完成该管理信息系统的数据库总体设计方案，明确数据库中表的结构，各表中关键字的设置，表与表之间的关系。

3．根据系统功能需求，以SQL语句的形式分类列出系统应涉及的数据操作。

4．选用熟悉的数据库工具，根据设计方案正确建立数据库，并成功实现上述数据操作。

5．完成上述内容，并提交书面实验报告。

1. **实验步骤：**

实验平台：windows10,IDEA2016,SQLServer2016

1. **建立关系模型，用ER图表示如下：**

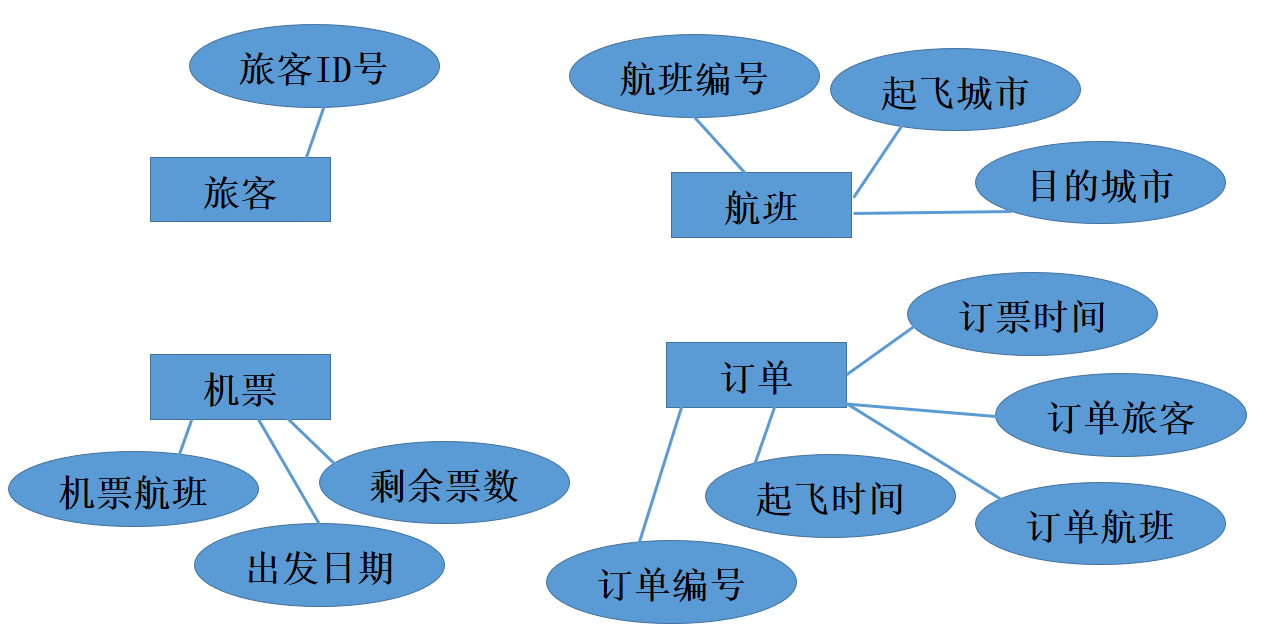


图3-1-1：实体ER图

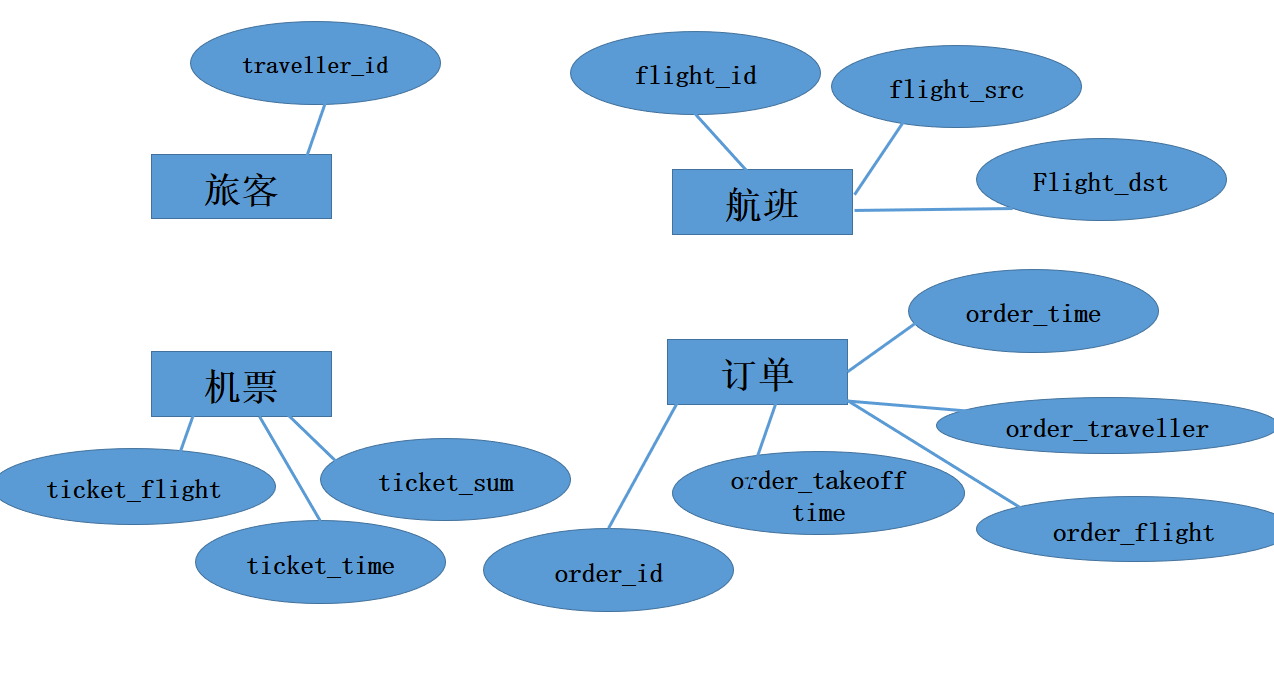
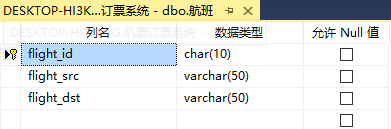


图3-1-2：ER图

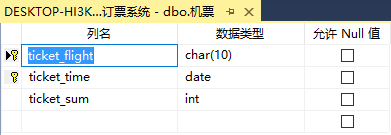
1. **定义数据库表**
2. 航班表



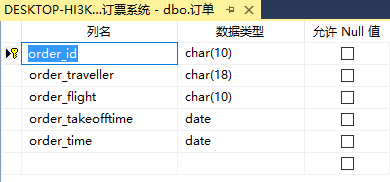
1. 旅客表



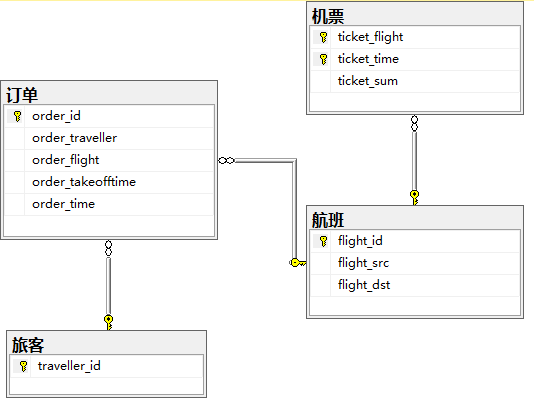
1. 机票表



1. 订单表



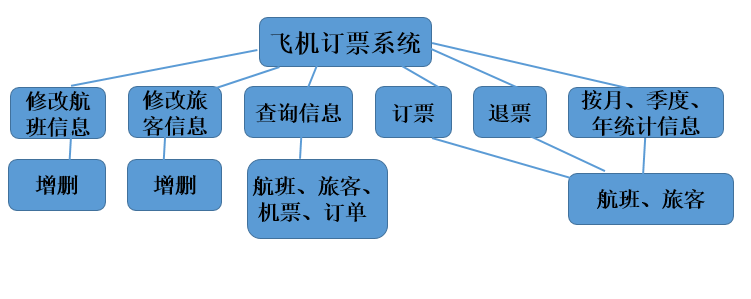
1. 机票订票系统的数据关系图：



在上述关系中，机票表中tiket\_flight为航班表中flight\_id的外键，删除航班表中特定航班时机票表中特定航班的所有机票被删除（级联删除）。订单表中order\_traveller为旅客表中traveller\_id的外键，当删除旅客信息时，订单表中所有该旅客的订单被删除。订单表中order\_flight为航班表中flight\_id的外键，当删除旅客信息时，订单表中所有该旅航班的订单被删除。

当旅客定票时，机票表中的剩余票数会减少，当旅客退票时，机票表中的剩余票数增加，从而实现动态刷新库存。订单表中的order\_time和order\_takeofftime是为了便于按月、按季度、按年份统计航班售票情况和旅客出行情况而设定的。

1. **软件功能实现**



1. **相关实现代码说明：**

**a.主函数里面使用switch结构，实现相关功能切换，不同命令执行不同的功能函数，当用户键入“quit”时，退出系统：**

public static void main(String[] args) {

connect\_database();//连接数据库

System.out.println("请输入对机票订票系统的操作：\n 1、flight(航班修改)；\n 2、traveller(旅客修改)；" +

"\n 3、query(查询航班，旅客等信息)；\n 4、count(统计售票出行)；\n " +

"5、order(预定机票)；\n 6、refund(退机票)；\n 7、quit(退出系统)；\n");

Scanner sc=new Scanner(System.in);//System.in代表标准输入，就是键盘输入

sc.useDelimiter("\n");//Scanner默认以空格作为分隔符，这里以回车作为分隔符

while(sc.hasNext())

{

int flag=0;

switch (sc.next())

{

case "flight":

flight();

break;

case "traveller":

traveller();

break;

case "query":

query();

break;

case "order":

order();

break;

case "refund":

refund();

break;

case "count":

count();

break;

case "quit":

System.out.println("成功退出系统！");

flag=1;

break;

default:

System.out.println("提示：请重新输入正确的主菜单命令！");

}

if (flag==1)//flag=1,则退出机票订票系统

break;

}

}

**b.连接数据库的操作：**

Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");//加载JDBC驱动

con= DriverManager.getConnection(url);//解析url,连接数据库

if (con!=null)

{

System.out.println("连接成功.");

}

else

{

System.out.println("连接失败！");

}

**C.预定机票的操作，将旅客、航班、出发日期等信息输入到订单中：**

String order\_id=getRandomString(10);

String SQL\_order="INSERT INTO 订单(order\_id,order\_traveller,order\_flight,order\_takeofftime,order\_time)" +

"VALUES('"+order\_id+"','"+arr[0]+"','"+arr[1]+"','"+arr[2]+"','"+arr[2]+"')";

sql\_exe\_manipulate(SQL\_order);

String SQL\_ticket\_minus="UPDATE 机票 SET ticket\_sum=ticket\_sum-1" +

"WHERE ticket\_flight='"+arr[1]+"' AND ticket\_time='"+arr[2]+"'";

sql\_exe\_manipulate(SQL\_ticket\_minus);

System.out.println("预订机票："+order\_id+"\n");

**d.sql\_exe\_manipulate()和getRandomString()分别执行数据库SQL**的更新，查询操作：

public static void sql\_exe\_manipulate(String SQL) {

try {

Statement statement = con.createStatement();

statement.executeUpdate(SQL);

}

catch (Exception e)

{

e.printStackTrace();

}

}

//生成指定长度的随机数

public static String getRandomString(int length) {

String base = "abcdefghijklmnopqrstuvwxyz0123456789";

Random random = new Random();//构造函数

StringBuffer sb = new StringBuffer();//StringBuffer对象改变的是自身，在进行字符串处理时不生成新的对象，所以在内存使用上要优先String

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

int number = random.nextInt(base.length());//生成[0,base.length)区间的整数

sb.append(base.charAt(number));

}

return sb.toString();

}

**e. 查询航班信息：**

public static void iflight() {

try

{

String SQL="SELECT flight\_id AS 航班编号,flight\_src AS 起飞城市,flight\_dst AS 目标城市 FROM 航班";

ResultSet resultSet=sql\_exe\_query(SQL);

System.out.println(" 航班编号 起飞城市 目的城市");

while (resultSet.next()) {

System.out.println(resultSet.getString(1) + " " + resultSet.getString(2)+" "

+ resultSet.getString(3));

}

}

catch (Exception e)

{

e.printStackTrace();

}

}

**f. 按季度统计旅客出行情况**

public static void count\_traveller\_season(String traveller\_id) {

System.out.println("请输入你要查询的季度，例：2016年春/夏/秋/冬“2016 spring/summer/autumn/winter”\n");

Scanner scanner = new Scanner(System.in);

scanner.useDelimiter("\n");

String season\_s = scanner.next();

String[] arr = season\_s.split(" ");

String time\_start = arr[0] + "-01-01";

String time\_stop = arr[0] + "-03-31";

switch (arr[1])

{

case "spring":

time\_start = arr[0] + "-01-01";

time\_stop = arr[0] + "-03-31";

break;

case "summer":

time\_start = arr[0] + "-04-01";

time\_stop = arr[0] + "-06-30";

break;

case "autumn":

time\_start = arr[0] + "-07-01";

time\_stop = arr[0] + "-09-30";

break;

case "winter":

time\_start = arr[0] + "-10-01";

time\_stop = arr[0] + "-12-31";

break;

}

try {

String SQL\_season = "SELECT COUNT(\*)" +

"FROM 订单" +

" WHERE order\_traveller='" + traveller\_id + "'" +

" AND order\_takeofftime >='" + time\_start + "' AND order\_takeofftime <='" +time\_stop + "'";

ResultSet season\_resultSet = sql\_exe\_query(SQL\_season);

while (season\_resultSet.next()) {

System.out.println(traveller\_id + "旅客" + arr[0] + "年" + arr[1] + "季度共计出行： "

+ season\_resultSet.getString(1) + " 次");

}

} catch (Exception e) {

e.printStackTrace();

}

}

1. **实验系统演示：**
2. 增加、删除航班：

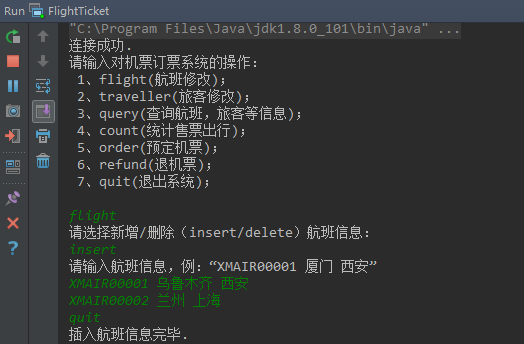
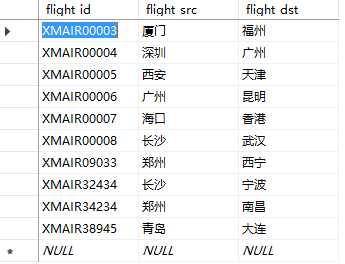
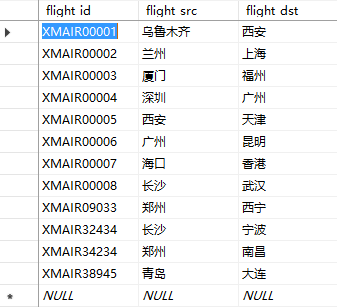


图4-1-1：插入航班的命令



图4-1-2：删除航班的命令



4-1-3:未插入航班前 4-1-4：新增航班后

1. 旅客注册：

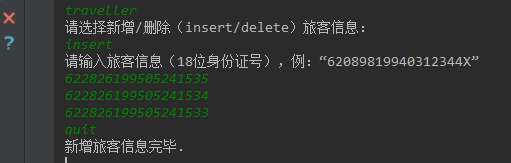


图4-2-1：增加旅客信息



图4-2-2：删除旅客信息

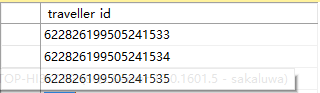


图4-2-3：增加旅客后的旅客表

1. 查询信息：

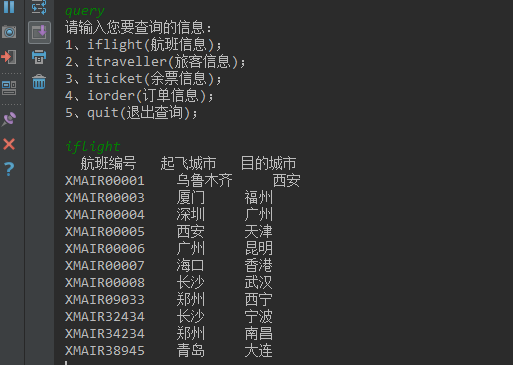


图4-3-1：查询航班信息

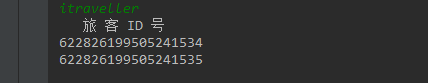


图4-3-2：查询旅客信息



图4-3-3：查询机票信息



图4-3-4:查询订单信息

1. 订票：



图4-4-1：机票预订成功

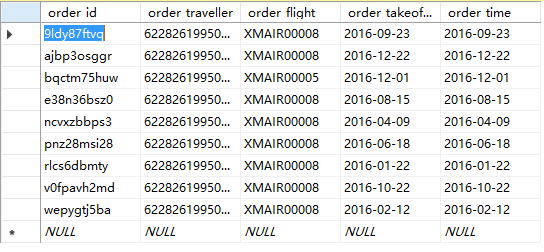


图4-4-2：预订机票生成的订单

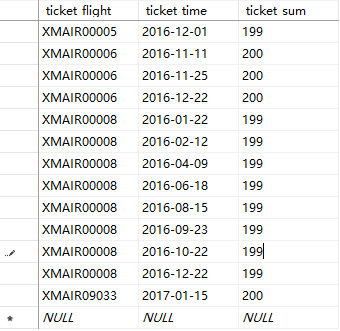


图4-4-3:机票库存动态刷新

1. 退票：

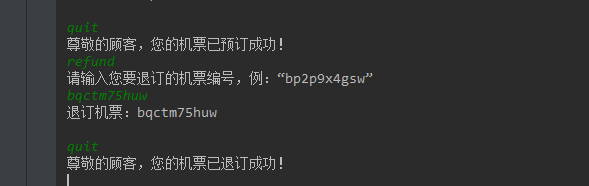


图4-5-1：退票

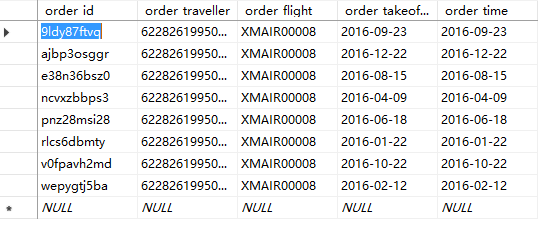


图4-5-2：相应订单被删除

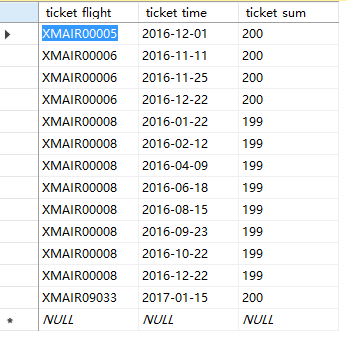


图4-5-3:机票库存被刷新

1. 统计航班售票情况：

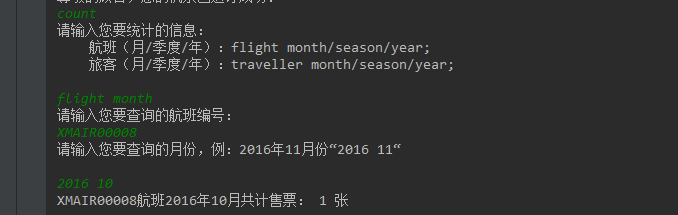


图4-6-1：航班XMAIR00008在2016年11月售票1张

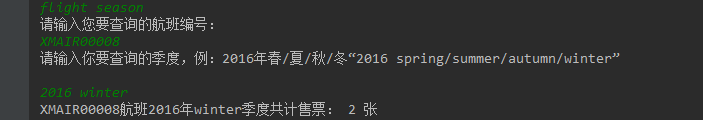


图4-6-2：航班XMAIR00008在2016年冬季售票2张

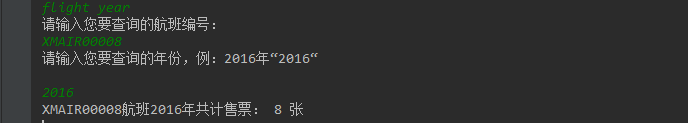


图4-6-3：航班XMAIR00008在2016年售票8张

1. 旅客出行情况统计：

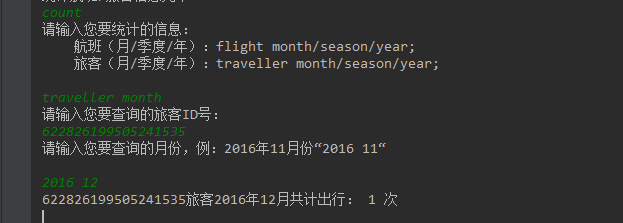


图4-7-1：旅客622826199505241535在2016年12月出行1次

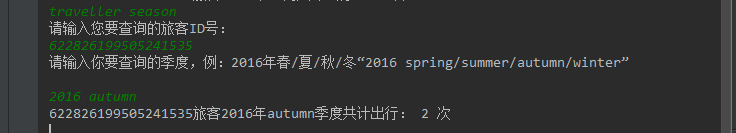


图4-7-2：旅客622826199505241535在2016年秋季出行2次

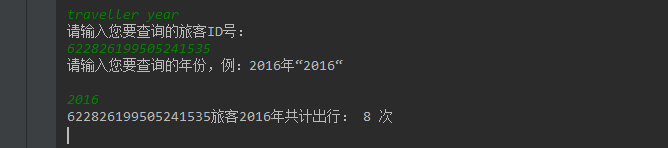


图4-7-3：旅客622826199505241535在2016年出行8次

1. 退出系统：

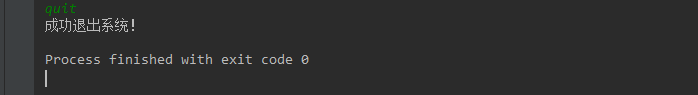


图4-8-1：quit命令退出机票订票系统

1. **实验总结：**
2. 以前的许多程序实验都是直接从程序中调用数据，用链表数组等将其存

储。像这样连接数据库系统将数据和程序如此分离的程序实验，我还是第一次做，不过其基本功能类似。

1. 本系统实现的功能有：航班信息的修改，旅客信息的修改，订票、退票

功能，信息查询功能，售票出行情况统计功能。

1. 该系统也存在很大的不足之处，由于我对于GUI编程并不是非常熟悉，

所以采用了命令行的方式来实现交互，使得系统不好用。另一方面，该系统的容错能力也非常有限，很多情况下并没有对用户输入做检查，有时候可能导致秩序崩溃。

1. 总之，这次数据库实验历时一周，数据库结构的设计、数据的填充，提

高了我系统设计的能力，软件功能的实现也让我的编程能力得到了锻炼。对于开发一个系统也有了初步的认识，以后还要多写程序多实践来提高自己的编程和设计能力；