河南工業大學

《数据库应用系统》

课程设计报告

题　　目：电脑维修管理系统的设计与实现

专 业 班 级 ：计科 F1602

学 生 姓 名 ：宋广辉

学　　号 ：201616010113

指 导 教 师 ：

课程设计时间：2018.12.29—2019.1.13

目录

1. 引言

随着计算机应用的日益普及和深化，每人一台电脑已经成为一种趋势。本项目要开发的是电脑维修管理系统。由于个人电脑的数量越来越多，原来的人工工作方式不仅会造成办理时间的延误和人力资源的浪费,特别是在维修的高峰期，很容易忘了，维修不及时，造成不必要麻烦。为此，我们决定开发本系统。

1.1 系统基本情况

本系统是一个小型的以电脑维修为主要业务的公司，主要对客户报修电脑管理、客户评价维修服务管理、内部维修回执管理、领配件的出入库信息管理等。系统开发的目标就是要准确、方便地对客户的电脑维修进行管理，及时、准确地将报修信息提供给维修人员进行是合理的维修服务。

现行系统是全手工管理，包括单正的制作和填制，全部都是由有关人员人手完成。电脑维修管理系统有三个目标，一、是管理电脑修理业务，二是管理回执和评价业务，三是管理配件和系统用户信息业务。

1.2 系统开发工具

开发系统的基本信息如下：

操作系统：Windows 10

集成开发工具：Eclipse

数据库：SQL Serve 2014

JDK：

服务器：Tomcat V9.0

浏览器：360 极速浏览器、Google Chrome 浏览器

版本控制工具：GitHub Desktop 和 Git

2. 系统需求描述

为了实现对电脑维修信息的统一管理，方便电脑维修店对所有维修的数据进行统计分析，本电脑维修管理系统面向电脑维修店的管理员和客户，对电脑报修、维修服务评价、更换的配件、故障类型等基本信息以及客户和管理员的信息等进行统一管理。本电脑维修管理系统根据面向使用者主要分为两个重要的子系统，包括面向客户（可以理解为用户）的客户子系统和面向管理员（可以理解为员工）的管理子系统。

2.1 业务需求描述

根据课程设计任务书的要求和相关描述，结合电脑维修管理的实际情况和自己对于电脑维修信息管理的理解，以及了解到的部分电脑店的运行管理情况。电脑维修管理系统主要应该提供如下功能。

（1） 客户基本信息的管理，包括提供客户信息的增加、查询、显示、修改和删除的基本功能，包括：

- 客户自行注册账户和密码，然后系统录入信息并为其分配唯一的身份编号
- 客户通过邮箱等信息申请获取自己的编号和登录名等信息
- 管理员登录后，修改客户的信息、查看客户的全部信息、删除客户的信息
- 密码信息的加密

（2） 管理员基本信息管理，包括管理员信息的增加、删除、修改、查找和显示等功能，包括：

- 管理员信息录入、维护和查询
- 管理员录入后，自动分配唯一的身份编号
- 密码信息的加密

（3） 维修信息管理，主要提供维修单的创建、删除、查询和显示以及修改，包括：

- 自动生成唯一的维修单号，不需手动录入
- 客户登陆后，可以创建维修单，并保存报修的时间
- 客户可以查看自己的维修单，还可以显示是否已评价该维修单
- 客户还可以查看系统所有的维修单

● 管理员可以删除和修改维修单的信息

（4） 评价信息管理，主要包括评价信息的创建、删除、查询和维护，包括：

● 客户可以在维修完成后，对维修服务做出评价

● 所有的客户可以看到系统所有的评价信息

● 客户还可以修改和删除自己的评价信息

● 管理员也可以看到系统所有的评价信息

● 管理员可以删除客户的评价信息，但是删除自己的维修单的评价

（5） 回执单信息管理，主要包括回执单的创建、删除、查询和维护，包括：

● 管理员填写回执单并记录管理员编号和时间

● 管理员可以修改、维护、查询回执单

● 回执单数量统计功能

（6） 配件信息管理，主要包括电脑配件信息的增加、删除、修改和查询以及更新库存和价格，包括：

● 管理员填写回执单后同步更新配件库的信息

● 电脑配件信息的录入、修改和查询以及删除

● 电脑配件信息的统计以及价格变动表

（7） 权限管理，主要提供权限分配、登录、登录验证和权限验证等功能，包括：

● 管理员权限分配和登录验证

● 用户登录、登录验证和权限验证

● 登陆后可以保存密码到 Cookie，便于下次登录

● 根据邮箱获取登录名以及申请管理员和用户的密码重置

（8） 信息统计和分析，主要提供系统中一些关键信息的统计和分析以及显示功能，包括：

● 系统时间的显示功能

● 维修单、评价单的信息统计和分析功能

● 客户数量和页面访问量的统计

根据以上功能描述，电脑维修管理系统的主要功能模块如图 2-1 所示：

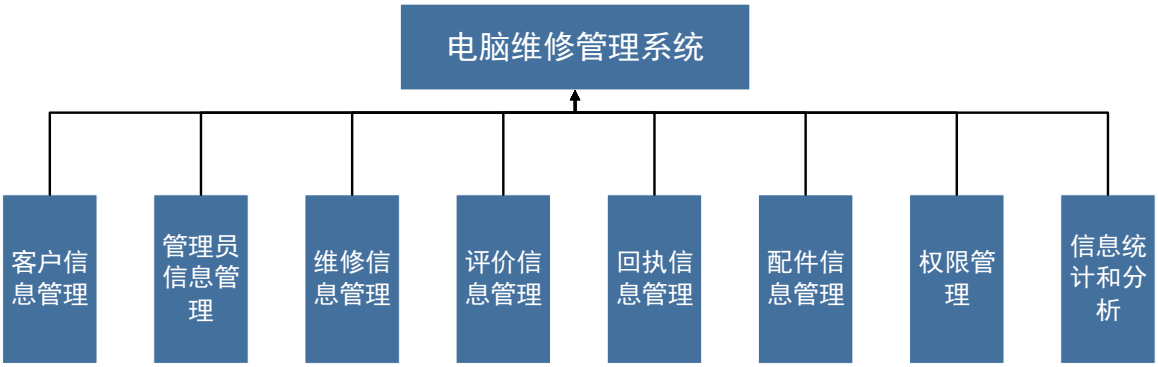


图 2-1　电脑维修管理系统主要功能模块图

2.2 数据需求描述

根据功能需求分析和描述，电脑维修系统的数据需求分析和描述如下：

（1）　客户注册信息：包括唯一标识符、客户登录名、密码、姓名、性别、出生日期、邮箱、电话、住址、注册日期、会员的积分数、单位等信息。系统检查所有的信息填写正确后提示客户注册成功，并返回到客户登录页面，同时对客户的密码进行加密，防止 SQL 数据库泄密造成客户密码泄露。客户登陆后，可以完成电脑报修、维修服务评价、查看维修记录和评价信息以及维护维修和评价信息。客户具有一定的积分数，可累加，一次维修并完成服务评价后，可以自动获得一个积分，同时如果更换了大型配件，也可以获得一定的积分。

（2）　管理员注册信息：包括唯一标识符、登录名、密码、姓名、性别、邮箱、电话、住址、注册日期、等级和上级管理员编号等信息。系统默认有一个 root 超级管理员，只能通过管理员才有权限申请管理员，同时对管理员的密码进行加密，防止 SQL 数据库泄密造成密码泄露。

（3）　维修单信息：包括维修单单号、客户 ID 号、客户申请报修时间、联系人、联系电话、联系地址、维修时间、故障所属类型、电脑问题描述、备注等信息。维修单单号是唯一标识，由系统根据一定的算法自动生成。维修时间表示客户期望的维修服务时间，不是维修完成时间，维修完成时间保存在回执单中，请注意区分。

（4）　客户评价信息：包括评价记录编号、维修单单号、客户 ID 号、维修员（员工）ID 号、评价时间、服务星级、评价的具体内容、备注等信息。评价记录编号是唯一标识符。评价时间不需要输入，在客户提交评价信息时自动生成，服务星级分为 1-5 级，用数字表示，5 代表满星级的优秀服务。

（5）　回执单信息：包括回执单编号、维修单号、维修员（管理员）编号、维

修员（管理员）维修完成时间、是否更换了配件、配件类型、故障解决方案、避免故障的建议、备注等信息。回执单编号为唯一标识符。维修单号需要管理员手动输入（或者从下拉列表中选择），维修员编号默认填写回执单的管理员 ID，维修完成时间默认为填写回执单的时间。

（6） 配件信息：包括编号，配件名称、配件类型、生产商、总量、余量、进价、售价、备注。编号是唯一标识符。

（7） 系统反馈信息：包括反馈编号、反馈者姓名、反馈者联系邮箱、反馈信息主题、反馈信息详细描述等信息。反馈编号是唯一标识符。反馈者姓名和反馈者邮箱是选填项，可以不填，另外除了收集上述信息外，不允许再收集其他信息。

2.3 业务规则及完整性约束描述

基于上述功能描述和数据需求描述，通过进一步地深入调查和了解以及和一些小型的电脑维修店沟通，电脑维修管理系统的业务规则及完整性约束描述如下：

（1） 所有进入系统的用户，无论其是游客、客户、管理员都可以对系统进行反馈，且不保存其个人信息。

（2） 客户和管理员在登录和注册时，需要校验账号（即登录名）应该是 3-10 个字母数字下划线组成，密码应该由 6-20 位字母数字下划线和点“.”组成；性别字段只能是‘男’或者‘女’，还有校验邮箱的输入是否合法。

（3） 注册用户、注册管理员、故障报修、对维修服务做出评价和确认回执单时均需要自动记录操作的时间和管理员或者客户的编号，便于分析和统计。

（4） 用户（客户）在报修时，需要填写或者选择期望的维修员提供维修服务的时间，以及联系人、联系电话和维修地址，并且期望的维修时间不得早于登记报修的时间。

（5） 客户评价信息，必须是在管理员填写该回执单后才可以评价，否则不允许评价此次维修服务。

（6） 填写回执单时，如果使用了配件，需要及时自动更新配件库相应配件的信息，且不允许在回执单中填写配件库中没有的配件，防止维修人员私下使用配件，既无法确保配件的质量，还会影响声誉。

（7） 所有客户均可以查看本系统所有客户的维修记录和所有的评价信息，但是必须是登录系统后的客户才可以查看，未登录的客户只能填写系统反馈或者注册

成为客户。

（8） 所有的管理员均可以注册下级管理员，但是需要记录其管理员编号，即记录每一个管理员的上级管理员 id 号，不可更改且不可以是它本身，但是超级管理员 root 例外，它的上级管理员仍是它自己。

（9） 一个客户可以进行多次报修，一次报修也可以被客户评价多次；一次报修只能有一个回执信息，一个管理员可以填写多次报修的回执信息；一次维修可以使用多个配件，也可以不使用配件。

（10） 客户报修之后，报修的单号应该是系统自动生成的，且是不可重复，于此同时，还应该保证在一定的并发量的条件下仍不会出错。

（11） 实验中所有的密码都应该是加密之后再存放到数据库中，且加密算法最好是不可逆的，即使数据库被黑客等非法窃取，也不能获得客户和管理员的密码，在一定程度上保证密码的安全性。

（12） 客户或者管理员登陆之后，若使用记住密码功能，则应允许浏览器在一定的时间段内记住其账号和密码，以便于之后快速登录和进入系统。

3. 系统设计

系统采用 B/S 模式的网站方案,利用 MVC 模式进行网站的设计和开发。MVC 全名是 Model View Controller，是模型(model)－视图(view)－控制器(controller)的缩写，一种软件设计典范，用一种业务逻辑、数据、界面显示分离的方法组织代码，将业务逻辑聚集到一个部件里面，在改进和个性化定制界面及用户交互的同时，不需要重新编写业务逻辑。基本 MVC 的思想，系统的设计主要可以分为数据库设计和网站设计。

3.1 数据库设计

数据库设计主要就是在需求分析的基础上，完成基本实体集及其属性的确定，根据基本实体集找到实体集与是集体之间的联系集联系属性，在此基础之上，进一步完善实体集与实体集之间的联系，画出完整的 E-R 图以及给出数据字典。

3.1.1 概念模型

综合以上分析和设计，可以给出电脑维修管理系统的完整 E-R 图，如图 3-2 所示。

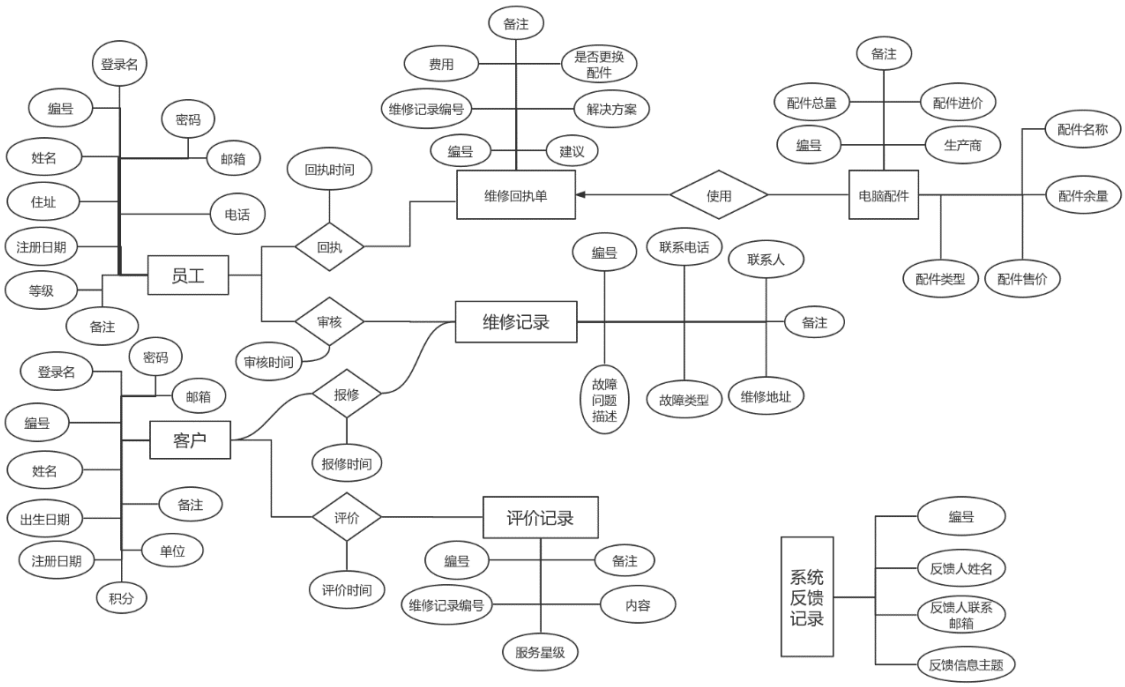


图 3-1 概念模型的完整 E-R 图

3.1.2 关系模式

将上述 E-R 图转化为关系模式和数据字典，可得到下面的关系模式和数据字典表如下：

（1）员工实体集转化的关系模式和数据字典如下：

admin(<u>id</u>，loginname，password，name，sex，email，phone，address，date，grade，upperid，*remark)*

表 3-1 员工（admin）实体集的数据字典

字段名称	数据类型	可否为空	字段描述	其他说明
id	int	否	唯一标识符	自增，主键
loginname	varchar(10)	否	员工登录名	
password	varchar(10)	否	密码	

name	varchar(20)	是	姓名	
sex	varchar(10)	否	性别	
email	varchar(20)	否	邮箱	
phone	varchar(20)	是	电话	
address	varchar(40)	是	住址	
date	datatime	否	注册日期	
grade	varchar(20)	是	等级	
upperid	int	否	上级管理员编号	
remark	text	是	备注	

（2）客户实体集转化的关系模式和数据字典如下：

custom(id,loginname,password,name,sex,email,phone,address,birthday, date, point, unit, remark)

表 3-2　客户（custom）实体集的数据字典

字段名称	数据类型	可否为空	字段描述	其他说明
id	int	否	唯一标识符	自增，主键
loginname	varchar(10)	否	客户登录名	
password	varchar(10)	否	密码	
name	varchar(20)	是	姓名	
sex	varchar(10)	否	性别	
email	varchar(20)	否	邮箱	
phone	varchar(20)	是	电话	
address	varchar(40)	是	住址	
birthday	datatime	是	出生日期	
date	datatime	否	注册日期	
point	int	是	会员的积分数	
unit	varchar(20)	是	单位	
remark	text	是	备注	

（3）维修记录实体集转化的关系模式和数据字典如下：

repair orders（orderID，*customID*，applyTime，linkName，linkPhone，

linkAdress，repairTime，problemType，problemDescription，remark）

表 3-3　维修记录（repair orders）实体集的数据字典

字段名称	数据类型	可否为空	字段描述	其他说明
orderID	varchar(25)	否	维修单单号	
customID	int	否	客户 ID 号	
applyTime	datetime	否	客户申请报修时间	
linkName	varchar(20)	否	联系人	
linkPhone	varchar(20)	否	联系电话	
linkAdress	varchar(20)	否	联系地址	
repairTime	datetime	否	维修时间	
problemType	varchar(20)	否	故障所属类型	
problemDescription	text	否	电脑问题描述	
remark	text	是	备注	

（4）评价记录实体集转化的关系模式和数据字典如下：

evaluation record（evaluateID，*orderID*，*userID*，*adminID*，evaluateDate，starLevel，Context，remark）

表 3-4　评价记录（evaluation record）实体集的数据字典

字段名称	数据类型	可否为空	字段描述	其他说明
evaluateID	int	否	评价记录编号	自增
orderID	varchar(25)	否	维修单单号	
userID	int	否	客户 ID 号	
adminID	int	否	维修员（员工）ID 号	
evaluateDate	datatime	是	评价时间	
starLevel	int	否	服务星级	
Context	text	否	评价的具体内容	
remark	text	是	备注	

（5）维修回执实体集转化的关系模式和数据字典如下：

repair receipt（receiptID，*orderID*，*adminID*，repairTime，isNeedParts，*partsType*，resolvent，advise，remark）

表 3-5　维修回执（repair receipt）实体集的数据字典

字段名称	数据类型	可否为空	字段描述	其他说明
receiptID	int	否	回执单编号	自增
orderID	varchar(25)	否	维修单号	
adminID	int	否	维修员（员工）编号	
repairTime	datetime	是	维修员（员工）维修完成时间	
isNeedParts	boolean	否	是否更换了配件	
partsType	varchar(20)	是	配件类型	
resolvent	text	否	故障解决方案	
advise	text	是	避免故障的建议	
remark	text	是	备注	

（6）系统反馈实体集转化的关系模式和数据字典如下：

feedback（ID，linkName，linkEmail，theme，description）

表 3-6　系统反馈（feedback）实体集的数据字典

字段名称	数据类型	可否为空	字段描述	其他说明
ID	int	否	反馈信息编号	自增
linkName	varchar(20)	是	反馈者姓名	
linkEmail	varchar(20)	是	反馈者联系邮箱	
theme	text	否	反馈信息主题	
description	text	否	反馈信息详细描述	

（7）配件信息实体集转化的关系模式和数据字典如下：

pc parts（id，name，type，producer，total，remainder，bid，price，remark）

表 3-7　配件信息（pc parts）实体集的数据字典

字段名称	数据类型	可否为空	字段描述	其他说明
id	int	否	配件编号	自增

name	varchar(20)	否	配件名字	
type	varchar(20)	否	配件类型	
producer	varchar(50)	否	配件生产商	
total	int	否	配件总量	
remainder	int	否	配件余量	
bid	float	否	配件进价	
price	float	否	配件售价	
remark	text	是	备注	

3.2 网站设计

网站设计主要分为前端设计和后端设计，本系统前端采用 JSP 技术、JavaScript 技术和 bootstrap 框架进行前端的开发，后端则利用 Servlet 技术进行前后端的联系、DAO 原理进行数据访问控制、Filter 过滤技术进行权限控制，利用 SQL Serve 提供基本的数据操作；同时采取代理模式和工厂进行操作的封装和解耦，提高系统可维护性。同时加密算法采取常用的 MD5 加密算法，在实现密码安全不可逆的同时，还可以减轻加密的难度降低服务器的压力，而自动生成维修单号算法则采取锁机制实现在一定并发量情况下随机生成唯一号码。

4. 系统实现

4.1 界面实现

（1） 客户登录页面：

图 4-1　客户登录页面

管理员登录页面、找回密码页面、反馈信息页面、注册客户页面、注册管理员页面等均于此页面类似。其关键代码为：

```
<%
Cookie cookies[]=request.getCookies();
String name="";
String password="";
if(cookies!=null){
   for(int i=0;i<cookies.length;i++){
      if(cookies[i].getName().equals("CustomCookie")){
         name = cookies[i].getValue().split("#")[0];
         password = cookies[i].getValue().split("#")[1];
      }
   }
}
%>
   <div class="container">
      <p class="title">唯 e 客户服务系统</p>
      <div class="box">
            <div id="login_box">
                  <h2>登录页面</h2>
                  <form action="Login" method="post" name="form"
onsubmit="return checklogin()">
                  <input type="hidden" name="action"
value="CustomLogin">
                  <div class="ui field">
                  账号：<input id="name" type="text" name="name"
value="<%=name %>"><br>
                  </div>
                  <div class="ui field">
                  密 码 ： <input id="password" type="password"
```

```
name="password"value="<%=password %>"><br>
                        </div>
                        <div class="ui check">
                        <input       id="checkbox"       type="checkbox"
name="autoLogin"  checked="checked"  value="save">  记 住
 密 码 <br>
                        </div>
                        <div class="m">
                            <input   class="ui   button"   type="submit"
value="登录">
                            <a  href="UserRegister.jsp"><input  class="ui
button" type="button"  value="注册"></a>
                            <a  href="AdminLogin.jsp"  class="ui  button">
管理员登录</a>
                        </div>
                        </form>
                </div>
                <br>
                <div><pre> 唯    e  客 户 服 务 系 统 - 登 录 页
面!   <a href="foundmm.jsp">找回密码</a></pre></div>
            </div>
        </div>
        <br>
        <p class="foot">
            © WeiyiNetClient v1.1.1
            <br>
        </p>

    @Override
        public List<Custom> findByName(String name) throws Exception {
            List<Custom> list = new ArrayList<Custom>();
            String sql = "select id from custom where loginname=? ";
              stat = con.prepareStatement(sql);
              stat.setString(1,name);
              ResultSet rs = stat.executeQuery();
              Custom vo=null;
              while(rs.next()){
                vo=new Custom();
                vo=this.findById(rs.getInt(1));
                    list.add(vo);
              }
              return list;
    }
```

（2） 客户报修页面：

图 4-2 客户主界面的客户报修页面

客户评价页面、管理员回执页面、录入配件信息、查看配件信息、查看所有评价、查看所有维修、查看所有回执等页面均于此页面类似，关键代码为：

```
<div id="header">
<div class="container">
   <ul class="nav nav-tabs nav-justified" >
      <li class="active"><a href="">主页</a></li>
     <li ><a href="http://www.sheensong.top/wordpress">论坛</a></li>
      <li ><a href="Feedback.jsp">反馈</a></li>
<%
String
name=(String)request.getSession().getAttribute("CustomSession");
%>
          <li class="disabled" ><a><%=name %>用户</a></li>
          <li ><a href="UserLogin.jsp">退出登录</a></li>
      </ul>
      <script src="./js/bootstrap.min.js"></script>
   </div>

<%@                                                    page
import="java.sql.*,java.util.*,weiyi.dao.*,weiyi.dao.vo.*,weiyi.dao.fact
ory.*" %>
<%
Cookie cookies[]=request.getCookies();
String loginId="";
int id=0;
String name="";
if(cookies!=null){
   for(int i=0;i<cookies.length;i++){
      if(cookies[i].getName().equals("CustomCookie")){
         name = cookies[i].getValue().split("#")[0];
```

```
                loginId = cookies[i].getValue().split("#")[2];
                id=Integer.parseInt(loginId);
            }
        }
    }
    %>
        <div class="container">
            <form  action="CustomServlet" method="post">
                <div class="form-group">
                    <input           type="hidden"           name="action"
value="CreateRepairOrders">
                    <input          type="hidden"          name="customId"
value=<%=loginId %>>
                    <label  style="font-size:  18px;font-weight:  50;text-
align: center;">编号是：<%=loginId %>  的客户，您好！</label>
                    <label  style="font-size:  18px;font-weight:  50;text-
align: center;">维修时间：</label>
                    <input    class="form-control    Wdate"    type="text"
style="color:     #0000AA;width:30%;height:2.3em"     name="repairTime"
onClick="WdatePicker({el:this,dateFmt:'yyyy-MM-dd HH:mm:ss'})" value="请
选择您希望的维修员服务时间！">
            <label  style="font-size:  18px;font-weight:  50;text-align:
center;">备注：</label>
            <input  class="form-control"  type="text"  style="width:50%"
name="remark">
            <input class="btn btn-primary  active" type="submit" value="
确认报修  ">
        </div>
    </form>
    </div>

    @Override
        public int doCreate(RepairOrders vo) throws Exception {
            String sql = "{call RepairOrdersInsert(?,?,?,?,?,?,?,?,?)}";
             stmt = con.prepareCall(sql);
             stmt.setString(1, vo.getOrderId());
             stmt.setInt(2, vo.getCustomId());
             stmt.setTimestamp(3, vo.getApplyTime());
             stmt.setString(4,vo.getLinkName());
             stmt.setString(5, vo.getLinkPhone());
             stmt.setString(6, vo.getLinkAddress());
             stmt.setTimestamp(7, vo.getRepairTime());
             stmt.setString(8,vo.getType());
             stmt.setString(9, vo.getDescription());
```

```
            stmt.setString(10, vo.getRemark());

            int result=stmt.executeUpdate();
            if(result > 0) {
                return result;
            }
            else {
                return -1;
            }
    }
```

（3） 修改管理员信息页面：

图 4-3 修改管理员信息页面

修改配件信息、修改客户信息等页面于此类似，其核心代码为如下：

```
    <%@ page import="java.sql.*,java.util.*,weiyi.dao.*,weiyi.dao.vo.*,weiyi.dao.factory.*" %>
    <%
        request.setCharacterEncoding("utf-8");
        response.setCharacterEncoding("utf-8");
        Cookie cookies[]=request.getCookies();
        IDAO<Admin,Integer> dao=DAOFactory.getAdmin();
        Admin vo=null;
        int id;
        if(cookies!=null){
            for(int i=0;i<cookies.length;i++){
                if(cookies[i].getName().equals("AdminCookie")){

    id=Integer.parseInt(cookies[i].getValue().split("#")[2]);
                    vo=dao.findById(id);
                }
```

```
            }
        }
    %>
    <form   action="AdminServlet"   method="post"   name="registerform"
onsubmit="return checkregister()">
            <input type="hidden" name="action" value="AdminUpdate">
            <input type="hidden" name="id" value="<%=vo.getId() %>">
            <div class="ui register">
            账        号  : <input   id="name"
type="text" name="loginName" value="<%=vo.getLoginName() %>"><br>
            </div>
            <div class="ui register">
            密      码 : <input  id="password"
type="password" name="password" ><br>
            </div>
            <div class="ui register">
            备      注 : <input  id="remark"
type="text" name="remark"value="<%=vo.getRemark() %>"><br>
            </div>
            <div class="m">
            <input class="ui register button" type="submit" value="确认修
改">
            </div>
    </form>

    @Override
       public int doUpdate(Admin vo) throws Exception {
          String       sql       =       "update       admin       set
loginname=?,password=?,name=?,sex=?,email=?,phone=?,address=?,date=?,gra
de=?,upperid=?,remark=? where id=?";
             stat = con.prepareStatement(sql);
             stat.setString(1, vo.getLoginName());
             stat.setString(2, vo.getPassword());
             stat.setString(3, vo.getName());
             stat.setString(4, vo.getSex());
             stat.setString(5, vo.getEmail());
             stat.setString(6, vo.getPhone());
             stat.setString(7, vo.getAddress());
             stat.setTimestamp(8, vo.getDate());
             stat.setString(9, vo.getGrade());
             stat.setInt(10, vo.getUpperId());
             stat.setString(11, vo.getRemark());
             stat.setInt(12, vo.getId());
             int update=stat.executeUpdate();
```

```
            if(update>0) {
              return update;
            }else{
              System.out.println("更新失败！-AdminDAOImpl.doUpdate()方法
");
              return -1;
            }
    }
```

4.2 核心算法实现

利用锁机制和计数器原理实现一定并发量情况下随机自动生成订单算法，其关键代码如下：

```
    public class MakeOrderNum {
        /**
         * 锁对象，可以为任意对象
         */
        private static Object lockObj = "lockerOrder";
        /**
         * 订单号生成计数器
         */
        private static long orderNumCount = 0L;
        /**
         * 每毫秒生成订单号数量最大值
         */
        private static int maxPerMSECSize=1000;

        // 最终生成的订单号
        static String finOrderNum = "";

        public void makeOrderNum() {

        }

        /**
         * 获取订单号
         * @return 订单号
         */
        public static String getOrderNumber() {
            try {
                synchronized (lockObj) {
                    // 取系统当前时间作为订单号变量前半部分，精确到毫秒
                    long        nowLong        =        Long.parseLong(new
```

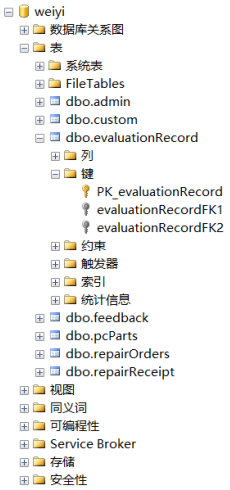
```
SimpleDateFormat("yyyyMMddHHmmssSSS").format(new Date()));
                    //  计数器到最大值归零，可扩展更大，目前 1 毫秒处理峰值
1000 个，1 秒 100 万
                    if (orderNumCount >= maxPerMSECSize) {
                        orderNumCount = 0L;
                    }

                    //组装订单号
                    String                                              countStr
=String.valueOf(maxPerMSECSize)+String.valueOf(orderNumCount);
                    finOrderNum=countStr +String.valueOf(nowLong);
                    orderNumCount++;
                    //System.out.println(finOrderNum      +      "--"      +
Thread.currentThread().getName() );
                    // Thread.sleep(1000);
                }
            } catch (Exception e) {
                e.printStackTrace();
            }

            //System.out.println(finOrderNum);
            return finOrderNum;
        }
}
```

4.3 数据库实现

（1） 创建表及主键和外键约束



（1） 创建 repairOrders 表的 SQL 脚本语句代码如下：

```
CREATE TABLE [dbo].[repairOrders](
    [orderid] [varchar](25) NOT NULL,
```

```
    [customid] [int] NOT NULL,
    [applytime] [datetime] NOT NULL,
    [linkname] [varchar](20) NOT NULL,
    [linkphone] [varchar](20) NOT NULL,
    [linkadress] [varchar](20) NOT NULL,
    [repairtime] [datetime] NOT NULL,
    [problemtype] [varchar](20) NOT NULL,
    [problemdescription] [text] NOT NULL,
    [remark] [text] NULL,
  CONSTRAINT [PK_repairOrders] PRIMARY KEY CLUSTERED
(
    [orderid] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS
= ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO
```

（2） 实现 RepairReceipt 表的主键和外键约束的 SQL 脚本如下：

```
alter table repairReceipt
    add constraint repairReceiptFK1 foreign key (orderid) references
repairOrders(orderid),
    constraint repairReceiptFK2 foreign key (adminid) references admin(id)
go
```

（3） 实现对 evaluationRecord 表的增删查改的存储过程的关键代码如下：

```
-- 插入的存储过程
CREATE PROCEDURE EvaluationRecordInsert
    @orderID    varchar(25)
    ,@userID    int
    ,@evaluatedate datetime
     ,@starlevel int
     ,@context     text
    ,@remark    text = NULL
AS
BEGIN
    INSERT INTO [dbo].[evaluationRecord](orderid
        ,userid
        ,evaluatedate
        ,starlevel
        ,context
        ,remark
```

```
    )
    VALUES(@orderID
    ,@userID
    ,@evaluatedate
     ,@starlevel
     ,@context
    ,@remark
    )
    return 1;
END
GO
-- 更新的存储过程
CREATE PROCEDURE EvaluationRecordUpdate
    @evaluateID int
    ,@orderID    varchar(25)
    ,@userID     int
    ,@evaluatedate datetime
     ,@starlevel int
     ,@context      text
    ,@remark     text = NULL
AS
BEGIN
    UPDATE [dbo].[evaluationRecord]
    SET orderid=@orderID
       ,userid=@userID
        ,evaluatedate=@evaluatedate
        ,starlevel=@starlevel
        ,context=@context
        ,remark=@remark
    WHERE evaluateid=@evaluateID
    RETURN 1;
END
GO
--创建删除的存储过程
CREATE PROCEDURE EvaluationRecordDelete
    @evaluateID int
AS
BEGIN
    DELETE FROM evaluationRecord
         WHERE evaluateid=@evaluateID

    RETURN 1;
END
```

```
GO
--创建查询单个记录的存储过程
CREATE PROCEDURE EvaluationRecordSelect(@evaluateID int)
AS
BEGIN
    SELECT * FROM evaluationRecord
        WHERE evaluateid=@evaluateID
END
GO


--创建查询所有记录的存储过程
CREATE PROCEDURE EvaluationRecordSelectALL
AS
BEGIN
    SELECT * FROM evaluationRecord
END
GO
```

（4） 实现自动获取订单号和实现自动获取已经回执的订单号的算法的关键代码如下：

```
-- 通过客户 ID 号查找其所有的订单编号
CREATE PROCEDURE OrdersId(@id int)
AS
BEGIN
    select orderid from repairOrders where customid=@id
END
GO
-- 通过客户 ID 号查找其所有已经填写回执的订单编号
CREATE PROCEDURE OrdersIdByReceipt(@id int)
AS
BEGIN
    select orderid from repairReceipt R
    where orderid in(select orderid from repairOrders O where customid=@id)
END
GO
```

5. 总结

两个星期的时间非常快就过去了，这两个星期不敢说自己有多大的进步，获得了多少知识，但起码是了解了一个 Java Web 项目开发的整个过程。通过这次课程设计发现这其中需要的很多知识我还没有接触过，去图书馆查资料的时候发现我们前边所学到的仅仅是皮毛，还有很多需要我们掌握的东西我们根本不知道。同时也发现有很多已经学过的东西我们没有理解到位，不能灵活运用于实际，不能很好的用来解决问题，这就需要我们不断的大量的实践，通过不断的自学，不断地发现问题，思考问题，进而解决问题。在这个过程中我们将深刻理解所学知识，同时也可以学到不少很实用的东西。

从各种文档的阅读到开始的需求分析、概念结构设计、逻辑结构设计、物理结构设计。亲身体验了一回系统的设计开发过程。很多东西书上写的很清楚，貌似看着也很简单，思路非常清晰。但真正需要自己想办法去设计一个系统的时候才发现其中的难度。经常做到后面突然就发现自己一开始的设计有问题，然后又回去翻工，在各种反复中不断完善自己的想法。

我相信有这样的问题不止我一个，事后想想是一开始着手做的时候下手过于轻快，或者说是根本不了解自己要做的这个系统是给谁用的。因为没有事先做过仔细的用户调查，不知道整个业务的流程，也不知道用户需要什么功能就忙着开发，这是作为设计开发人员需要特别警惕避免的，不然会给后来的工作带来很大的麻烦，甚至可能会需要全盘推倒重来。所以以后的课程设计一定要特别注意这一块的设计。

我做的是电脑维修系统。在需求分析过程中，我通过上网查资料，去图书馆查阅相关资料，结合我们的生活经验，根据可行性研究的结果和客户的要求，分析现有情况及问题，采用 B/S 结构，将电脑维修系统划分为两个子系统：管理员子系统，客户子系统。在两周的时间里，不断地对程序及各模块进行修改、编译、调试、运行，其间遇到了很多问题。

我们学习并应用了 SQL 语言，对数据库的创建、修改、删除方法有了一定的了解，通过导入表和删除表、更改表学会了对于表的一些操作，为了建立一个关系数据库信息管理系统，必须得经过系统调研、需求分析、概念设计、逻辑设计、物理设计、系统调试、维护以及系统评价的一般过程，同时学会了利用存储过程将数据库的一些操作进行封装，实现一定程度上的安全性保护。

很多事情不是想象中的那么简单的，它涉及到的各种实体、属性、数据流程、数据处理等等。很多时候感觉后面的设计根本无法继续，感觉自己的思路像是被前面做的各种图给限制了。在关系模型转换的时候碰到有些实体即可以认为是实体又可以作为属性，为了避免冗余，尽量按照属性处理了。最后，我做完 E-R 图的时候，又专门找了孙老师帮忙指点下 E-R 图中的，在他的帮助下，我又把图进行了完善。

虽然过程辛苦是不可避免，但收获还是令人感到尤其的欣慰。在这次的课程设计中不仅检验了我所学习的知识，也培养了我的实践能力，让我知道遇到一个问题，如何去寻找思路，如何去解决问题，最终完成整个事情。

最后，分享一个 GitHub 客户端解决中文乱码问题的经验，将所有文件的编码格式修改为 Utf-8 无 BOM 编码格式然后重新推送到 GitHub 远程即可解决，这里推荐直接将使用的集成开发工具（例如：Eclipse）的默认编码格式修改为 UTF-8，然后重新建立项目，把所有的代码复制过去保存，再用 GitHub 更新一下即可。

课程设计是我们专业课程知识综合应用的实践训练，是我们迈向社会，从事职业工作前一个必不少的过程。实验过程中，也十分感谢实验指导老师孙宜贵孙老师和唐建国唐老师的指点与教导。这次课程设计不仅是对这学期所学知识的一种综合检验，而且也是对自己动手能力的一种提高，增强了自己实践能力。通过这次课程设计使我明白了自己知识还比较欠缺，只是学习书本知识还是远远不够的，自己不会的东西还有太多，学习需要自己长期的积累，在以后的学习、工作中都应该不断的学习，将课本的理论知识与生活中的实践知识相结合，不断提高自己文化知识和实践能力。

参考文献

[1] 万常选，数据库系统原理与设计(第 3 版)，清华大学出版社，2017

[2] 耿祥义，Java 2 实用教程（第 5 版），清华大学出版社，2017

[3] QST 青软实训，Java 8 高级应用与开发，清华大学出版社，2016

[4] [英]托马斯;康诺利（Thomas Connolly），数据库系统：设计、实现与管理（基础篇）（原书第 6 版），机械工业出版社，2016

[5] [美]Ian F.Darwin，Java 经典实例，中国电力出版社

[6] 明日科技，Java Web 从入门到精通（第 2 版），清华大学出版社，2017