

黑龙江科技大学

Heilongjiang University of Science and Technology

专业综合训练 1(struts) 设计报告

设计题目: 移动计费账务管理系统

姓 名: 艾程

班 级: 软件工程 16-4 班

学 号: 2016024427 (08)

指导教师: 马晓梅

2018 年 12 月 28 号

综合成绩评定

考核项目		分值	A	C	得分
设计态度 (共 10 分)		10	按要求出勤，设计态度认真、积极	设计态度比较认真	
设计情况 (共 50 分)	设计工作量与难度	10	设计工作量达到要求、设计有一定难度	设计工作量与难度一般，基本达到了要求	
	设计方案	10	设计方案正确、合理	设计方案较正确、基本合理，但不是最优	
	设计完成情况	30	完成了选题的设计内容，设计功能完整，相关算法设计正确，程序结果正确、直观性好	基本完成了选题的设计内容及主要选题功能，相关算法设计基本正确，程序结果正确	
设计报告 (共 20 分)	报告组织结构及内容	10	内容组织及结构合理、内容充实、层次清晰、图表得当	内容组织及结构较合理、内容较充实、层次较清晰、图表应用基本得当	
	报告排版格式	10	格式规范，完全符合要求	格式基本规范，基本符合要求	
答辩情况 (共 20 分)		20	答辩思路清晰，问题回答准确	问题基本回答准确	
综合得分					
其它说明					
指导教师签名					
			年 月 日		

目 录

1 总体设计.....	2
1.1 设计简介	2
1.2 功能模块及分析.....	2
1.2.1 总体功能模块	3
1.2.2 具体功能模块	4
1.3 数据库设计.....	4
1.3.1 数据库环境说明	4
1.3.2 逻辑设计	5
1.3.3 物理设计	5
2 详细设计.....	6
2.1 程序系统的结构.....	6
2.2 程序设计说明.....	7
2.2.1 基础类.....	7
2.2.2 DB 层如何使用 JDBC 完成 DB 层的功能	7
2.2.3 显示层类.....	8
2.2.4 管理层类	9
2.2.5 Util 类.....	10
3 运行效果.....	12
4 结论	14

1 总体设计

1.1 设计简介

本项目是以 Struts 框架和 Mybatis 框架为基础，制作的移动公司的计费账务业务管理系统。

技术上不仅包含了页面布局 jsp 代码、后台流程逻辑 java 代码，还使用了 sql 语言操作数据库。

项目的主要功能是为移动的后台员工提供一个方便的修改客户信息，查看、管理后台人员的后台渠道。

本系统角色如下：

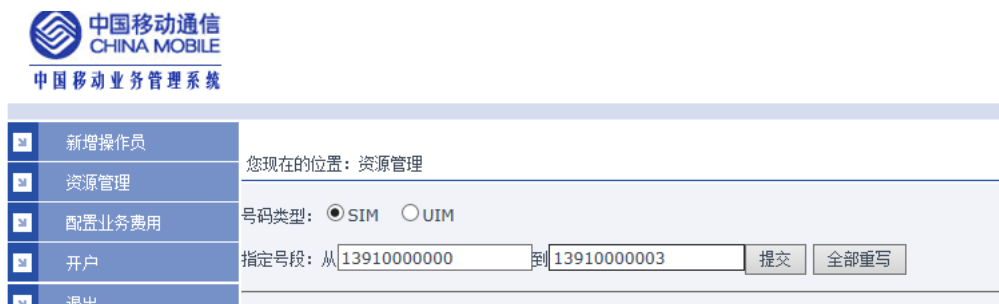
普通操作员：可以登录、新增客户信息、新增账户信息、退出注销。

管理员：可以登录、资源管理、配置业务费用、新增客户、新增用户、退出注销，还有一个重要的权限——升级/添加管理员。

1.2 功能模块及分析

本项目一共分为六大模块：登陆、新增操作员、资源管理、配置业务费用、开户、退出功能模块。

而我在项目中负责资源管理功能模块，具体页面如下：



这个模块的功能是将将号码存进数据库。

如图所示，点击“提交”，则会将类型为 SIM 卡，号段从 13910000000 到 13910000003 的号码存进数据库中。如果号码类型、指定号段没有选择数据，由于数据不能为空，会出现提交失败状况。

如果点击“全部重写”，则刷新该页面，默认的号码类型为 SIM 卡，并且清空指定号段一栏。

乔阳奇：系统登录和开户模块

史连杰：配置业务管理和系统退出模块

付佳宁：操作员管理模块

1.2.1 总体功能模块

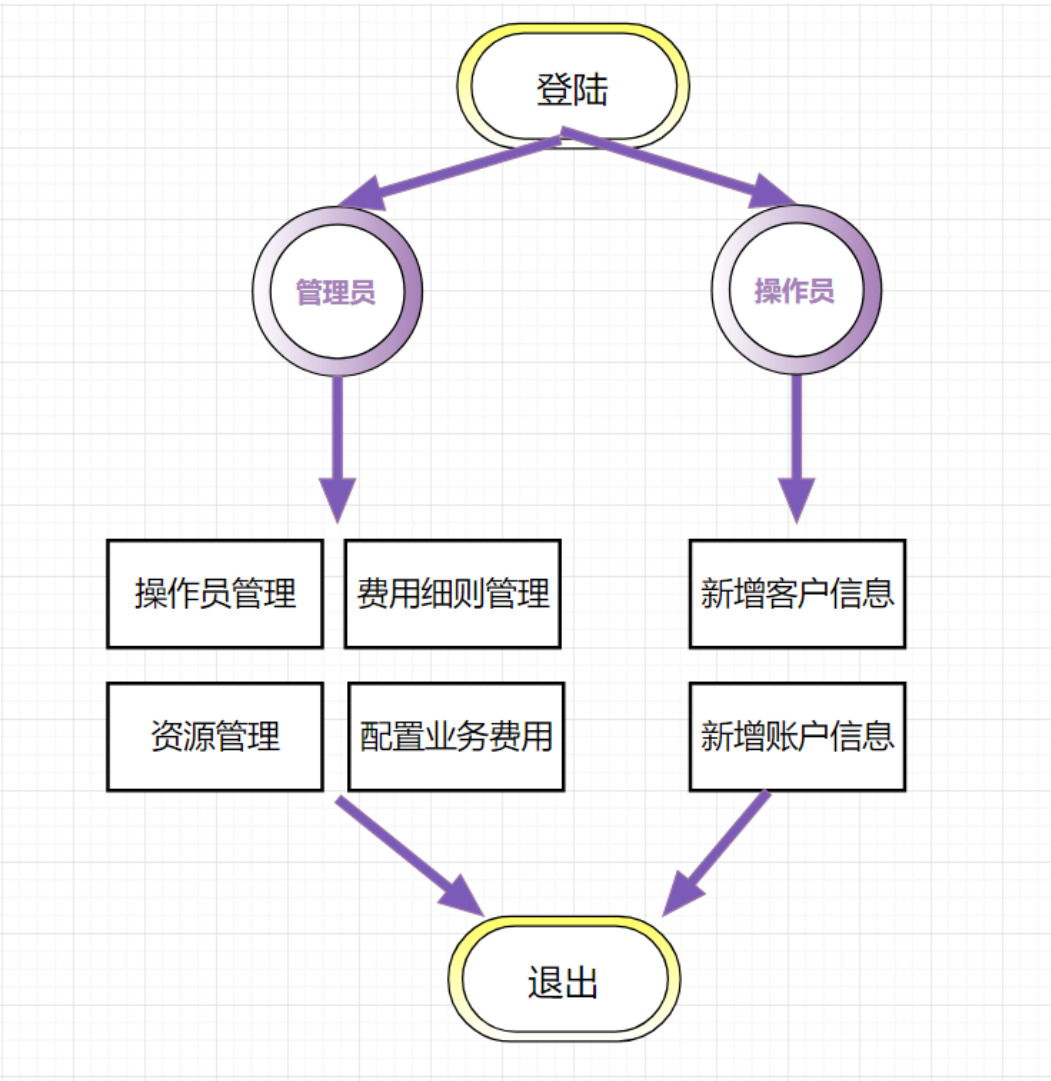


图 1 移动业务管理系统总体功能模块图

1.2.2 具体功能模块

当我们在页面中填写好了号码与选择类型后，可以选择“提交”、“全部重写”按钮。

点击“提交”，则会将类型、号段区间的号码存进数据库中。如果号码类型、指定号段没有选择数据，由于数据不能为空，会出现提交失败状况，这时候就会自动进入“全部重写”的逻辑流程。

如果点击“全部重写”，则刷新该页面，默认的号码类型为SIM卡，并且清空指定号段一栏。

1.3 数据库设计

1.3.1 数据库环境说明

数据库就是存放数据、管理数据的地方。

本项目的数据库采用了MySQL数据库，使用navicat作为管理数据库的软件：

使用mysql的原因有：

1. 开源，成本低。
2. 操作语句简单，容易实现。
3. 占用内存小，支持多种操作系统。
4. 运行速度快。
5. 支持常见的SQL语句规范。
6. 调试、管理、优化简单。

使用navicat作为管理数据库的软件原因是：

Navicat是一套快速、可靠并价格相当便宜的数据库管理工具，专为简化数据库的管理及降低系统管理成本而设。它的设计符合数据库管理员、开

发人员及中小企业的需要。Navicat 是以直觉化的图形用户界面而建的，我们可以以安全并且简单的方式创建、组织、访问并共用信息。

1.3.2 逻辑设计

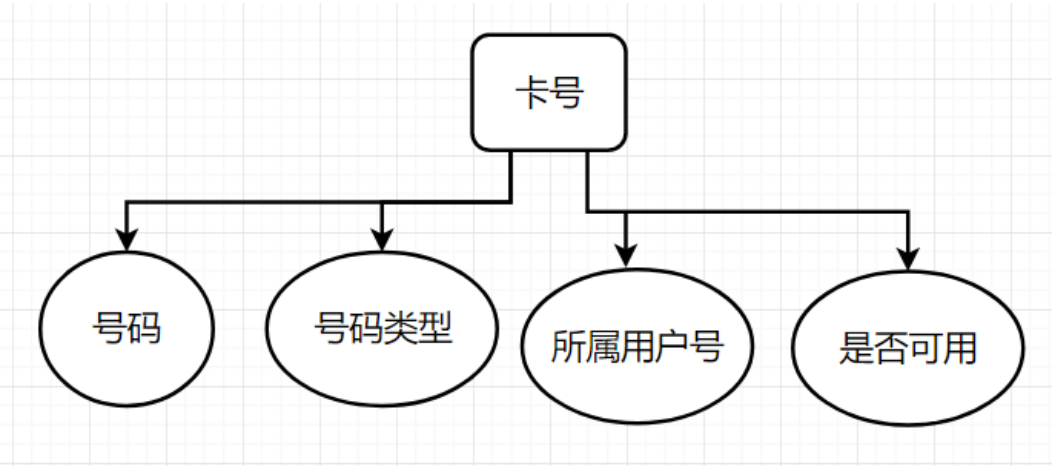


图 2 卡号资源管理 ER 图

1.3.3 物理设计

1. tmobiles 表

表 1 号码资源管理表

字段	类型	长度	备注
moblie_number	Int	16	号段
mobile_type	Varchar	10	号码类型
card_number	Int	16	所属用户号
is_available	Varchar	4	该卡号是否注销

2 详细设计

2.1 程序系统的结构

根据分析可以得到如图 2 所示的程序流程图。

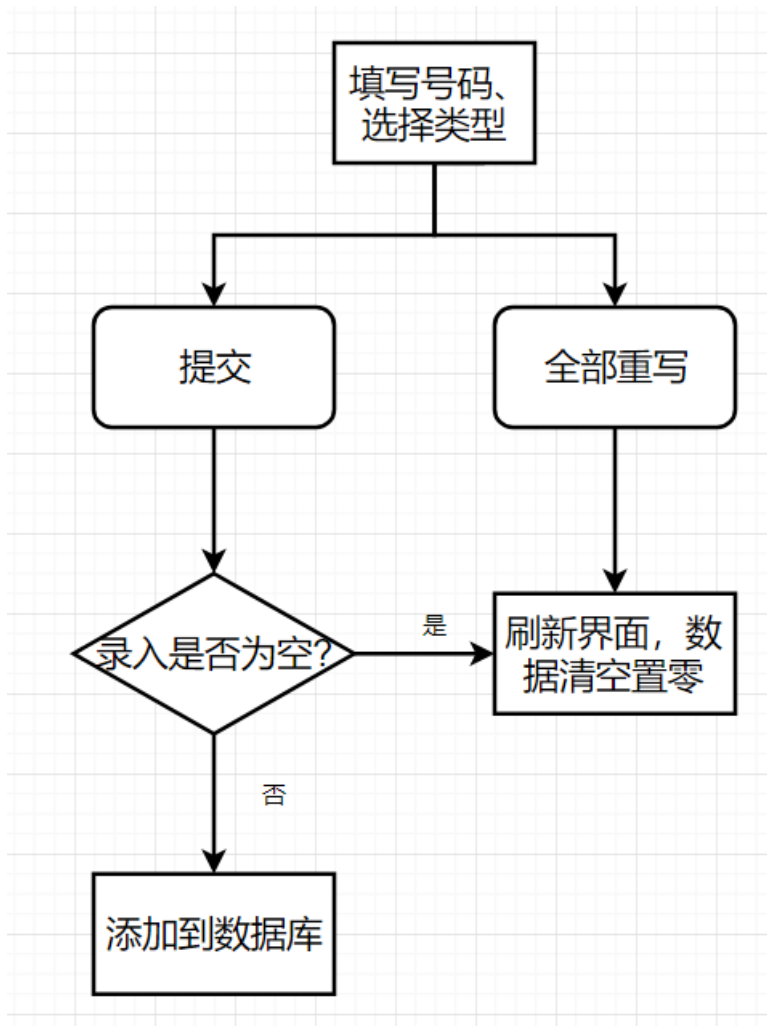


图 2 资源管理功能模块图

2.2 程序设计说明

2.2.1 基础类

我做的资源管理功能模块的核心 javabean 类为：

Mobiles.java （com.yidong.action 包中）

表 2 mobiles. java 属性

属性名	类型	说明
moblie_number	int	号码段
mobile_type	String	号码类型
card_number	int	所属用户号
is_available	String	是否可用

2.2.2 DB 层如何使用 JDBC 完成 DB 层的功能

由于该项目结合了 mybatis 框架和 struts2 框架。因此需要导入相应的 jar 包，DB 层使用的是 mybatis 框架。

为什么使用 mybatis 连接数据库？因为 mybatis 具有如下特点：

1. 与 JDBC 相比，减少了 50%以上的代码量。
2. 通过 MyBatis 操作数据库相当灵活，不会对应用程序或者数据库的现有设计强加任何影响，SQL 写在 XML 里，从程序代码中彻底分离，降低耦合度，便于统一管理和优化，并可重用。
3. 提供 XML 标签，支持编写动态 SQL 语句。
4. 提供映射标签，支持对象与数据库的 ORM 字段关系映射。

连接过程是：

在根目录 src 下配置 mybatisconfig.xml 文件。文件中主要包含需要连接到的数据库的用户名密码以及数据库名称和配置项目中需要用到的 mapper 映射文件。

通过该配置文件可以连接到需要操作的数据库表。

操作过程是：

- 1、新建相应数据表的 Dao 类的接口。
- 2、新建与接口相映射的 mapper.xml 文件。

2.2.3 显示层类

```

<form method="POST" action="addNum">
  <p>号码类型: <input type="Radio" name="MType" value="SIM" checked>SIM
  &nbsp;&nbsp;&nbsp;<input type="Radio" name="MType" value="UIM">UIM</p>
  <p>指定号段: 从<input type="text" name="T1" size="20" value="13910000000">
  到<input type="text" name="T2" size="20" value="13912345678"><input
  type="submit" value="提交" name="B1">
  <input type="reset" value="全部重写" name="B2"></p>
</form>
<hr>
</TD>
</TR>
</TBODY></TABLE>
<BR></CENTER></BODY></HTML>

```

2.2.4 管理层类

TMobiles. java 代码如下:

```

public class TMobiles {
    private String MOBILE_NUMBER;
    private String MOBILE_TYPE;
    private String CARD_NUMBER;
    private String IS_AVAILABLE;
    public String getMOBILE_NUMBER() {
        return MOBILE_NUMBER;
    }
    public void setMOBILE_NUMBER(String mOBILE_NUMBER) {
        MOBILE_NUMBER = mOBILE_NUMBER;
    }
    public String getMOBILE_TYPE() {
        return MOBILE_TYPE;
    }
    public void setMOBILE_TYPE(String mOBILE_TYPE) {
        MOBILE_TYPE = mOBILE_TYPE;
    }
    public String getCARD_NUMBER() {
        return CARD_NUMBER;
    }
    public void setCARD_NUMBER(String cARD_NUMBER) {

```

```

        CARD_NUMBER = cCARD_NUMBER;
    }
    public String getIS_AVAILABLE() {
        return IS_AVAILABLE;
    }
    public void setIS_AVAILABLE(String iS_AVAILABLE) {
        IS_AVAILABLE = iS_AVAILABLE;
    }
    @Override
    public String toString() {
        //为了方便调试，后台打印出来
        return "TMobiles [MOBILE_NUMBER=" + MOBILE_NUMBER + ", MOBILE_TYPE="
+ MOBILE_TYPE + ", CARD_NUMBER=" + CARD_NUMBER
        + ", IS_AVAILABLE=" + IS_AVAILABLE + "];"
    }
}

```

2.2.5 Uiti 类

由于调用次数繁多，为了减少代码量，减轻内存创建新对象的负担以及实现代码的可维护性。

本项目中使用了一个公用的接口类即 OperatorMapper。这样也避免在几乎每个 Action 中为了调用能操作数据库的方法而重复创建 sqlsession。代码如下：

```

public interface OperatorMapper {
    //查找所有管理员
    public abstract List<TOperator> findAllOperator();
    //通过名字找管理员
    public abstract TOperator findOperatorByName(String name);
    //添加操作员
    public abstract void addOperator(TOperator toperator);
    //添加号码字段
    public abstract void insertNumByTtoT(TMobiles t);
    //添加收费细项

```

```
public abstract void insertChargeItem(TCharge tc);
//查找所收费细项
public abstract List<String> findAllChargeCode();
//增加开户用户 TCustomer 表的信息
public abstract void insertCustomer(TCustomer tt);

public abstract void updateCustomer(TCustomer tl);

public abstract TCustomer findCustomerByNum(String num);

public abstract String findOperctorAdminByName(String name);

public abstract void updateOper(TOperator t);

public abstract String isAvailableNum(String num);

public abstract void addUser(TUser tuser);

public abstract void changeAvailableNum(String num);

public abstract void addChargeRule(TChargeRule tcr);

public abstract String findChargeCodeByName(String name);

public abstract void updateChargeByCode(TCharge tc);

public abstract void deleteChargeRuleByName(String name);

public abstract List<TChargeRule> findAllChargeRule();
}
```

3 运行效果

1、首先进入首页

路径: <http://localhost:8080/YiDongProject/index.jsp>:



图 3 项目主页图

2、输入用户名以及密码后, 点击“进入系统”后:

路径: <http://localhost:8080/YiDongProject/main>



图 4 项目后台页面图

3、点击“资源管理”后：

路径：<http://localhost:8080/YiDongProject/resource.jsp>

图 5 资源管理页面图

4、填写好号码类型以及字段后，点击“提交”，提交成功后会有相应提示：

路径：<http://localhost:8080/YiDongProject/resource.jsp>

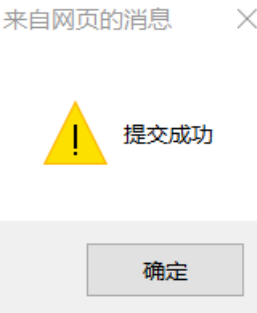


图 6 提交成功页面图

5、提交失败，或者点击“全部重写”后：

路径：<http://localhost:8080/YiDongProject/resource.jsp>

图 6 提交失败、全部重写页面图

4 总结

由于以前课设都是单人项目，本次课设是组队的项目，通过本次课设，我深刻了解了互相合作配合的流程以及重要性。并且也在本次课设中再次学习到了很多。

加深了课堂上曾经学到过的知识：

- 1、复习了利用 struts2 实现用户页面和数据库的数据传输与交换的逻辑代码。
- 2、连接数据库，并实现 CURD 操作。
- 3、复习了 js 代码的书写。
- 4、加深了对空指针异常的处理方法——找有涉及到数据传输的代码，或者配置出现了问题。

也同时学到了新的知识：

- 1、学会使用 mybatis 框架的连接、配置、基础流程操作。
- 2、了解了什么是映射，以及 mybatis 中的映射原理。
- 3、学会如何结合 mybatis 与 struts2 框架开发项目。
- 4、学会如何利用 MVC 模型分层，并且分工合作。

团队合作的过程中，我们团队是由组长发派任务，在配置好环境、框架的基础下，然后组员分工完成自己负责模块的任务，将自己跑通的代码上交，最后再由乔阳奇、史连杰将所有代码组合联系起来。

本次项目由于规模不大，因此没有“需求说明书”，但是我们团队之所以也能顺利完成项目，靠的是不断的交流与讨论。因此齐心协力才是项目成功能够成功开发出来的关键要素之一啊。