

停车场管理系统设计实验报告

【实验目的】

用JavaEE+MYSQL实现。JSP用于页面的设计，Java用来处理后台跳转、对数据库操作，实现对数据库的增、删、改等操作。使用Tomcat8.0作为WEB服务器，提供适应性强的Internet服务器功能，最终完成该项目

【实现功能】

该系统主要是实现以下功能：

系统信息管理：包括添加角色、管理角色、添加用户、管理用户、管理角色和管理用户模块可以对角色和用户进行增、删、改。

IC卡信息管理：添加IC卡类型，管理IC卡类型。管理IC卡信息可以对车主的IC卡信息进行增、删、改。

车位信息管理：管理车位、添加车位，管理车位信息模块可以对车主车位信息进行增、删、改。

固定车主停车管理：出入场设置、停车场信息管理。出入场设置可以设用户的入场和出场，相应的在停车信息中有一个展现。停车信息管理可以对车主的停车信息进行查询和删除。

临时车辆信息管理：车主入场信息，车主出场设置。设置车主入场，在相应的表中展现，在车主出场时进行收费，临时车主停车信息管理可以对临时车主的信息进行删除和查询。

收费管理：管理收费、添加收费信息。

系统功能操作：修改密码、退出系统。

【实验环境】

LINUX 系统或者虚拟机

Java 配置：JDK1.8

java 开发工具：Eclipse4.6

web 服务器：Tomcat8.0

环境配置过程：安装好JDK1.8后，选择Eclipse4.6的
Project->Web Project, 构建工程。

【实验内容】

登陆界面



部分代码：

```
publicclass Login {  
    publicboolean checkLogin(String user_id,String user_pwd)  
    {  
        String sqlCmd="select count(*) from user where user_id=? and user_pwd=?";  
  
        Object[] objList=new Object[2];
```

```

        objList[0]=user_id;
        objList[1]=user_pwd;
        String result=SQLUtil.excuteScalar(sqlCmd,objList).toString();
        if(result.equals("1"))
        {
            returntrue;
        }
        else {
            returnfalse;
        }
    }
    public String  getName(String user_id)
    {
        String sqlCmd="select user_name from user where user_id='"+user_id+"'";
        String result=SQLUtil.excuteScalar(sqlCmd, null).toString();
        return result;
    }
    public String  getSysLevel(String user_id)
    {
        String sqlCmd="select role_id from user where user_id='"+user_id+"'";
        String result=SQLUtil.excuteScalar(sqlCmd, null).toString();

        return result;
    }
}

```



系统管理界面

系统车位信息管理界面

部分代码

```
public class Seat {  
    public List<Object> getEntity()  
    {  
        String sqlCmd="select *from Seat";  
        return DBUtil.SQLUtil.executeQuery(sqlCmd, null);  
    }  
    public List<Object> getNoUseSeat()  
    {  
        String sqlCmd="SELECT *FROM Seat WHERE seat_id NOT IN(SELECT seat_id  
FROM card)";  
        return DBUtil.SQLUtil.executeQuery(sqlCmd, null);  
    }  
    public List<Object> getEntity(int page)  
    {  
        int size=(page-1)*15;  
        String sqlCmd="select *from Seat limit "+size+",15";  
        return DBUtil.SQLUtil.executeQuery(sqlCmd, null);  
    }  
    public List<Object> getEntityByWhere(String sqlWhere,int page)  
    {  
        int size=(page-1)*15;  
        String sqlCmd="select *from Seat where "+sqlWhere+" limit "+  
size+",15";  
        return DBUtil.SQLUtil.executeQuery(sqlCmd, null);  
    }  
}
```

系统信息管理	查询条件：角色编号 查询值： <input type="text"/> 查询
添加角色信息	
管理角色信息	
添加用户信息	
管理用户信息	
车位信息管理	

角色编号	角色名称	操作
z001	管理员	编辑 删除
z002	超级管理员	编辑 删除
z003	游客	编辑 删除

共 1 页 跳转至 转

```
public int deleteEntity(String seat_id)  
{  
    String sqlCmd="delete from Seat where seat_id='"+seat_id+"'";  
    return DBUtil.SQLUtil.executeUpdate(sqlCmd, null);  
}
```

```
public List<Object> getEntityById(String seat_id)
{
    String sqlCmd="select *From Seat where seat_id='"+seat_id+"'";
    return DBUtil.SQLUtil.executeQuery(sqlCmd, null);
}

public int updateEntity(String seat_id,String seat_num,String
```

数据库表格内容

根据系统需求确定系统有如下 7 张表。

用户表：储存用户信息。

角色表：储存角色信息。

车位管理表：管理车位信息以及车位状态。

IC 卡管理表：管理 IC 卡信息。

临时车辆记录表：记录临时车辆的信息。

固定车主表：记录固定车主信息及出入场信息。

收费表：记录用户们的收费信息。

用户表用于记录用户的编号、姓名、电话等属性信息

【实验感想】

本次停车场管理系统能够满足用户的基本需求，方便快捷，既节省了时间，提高了效率，又节约了开发成本，总体而言很满意，但是制作过程中还是出现了很多小问题，希望下次改正