第一章　引言

就目前来讲，伴随 Internet 旳迅猛发展，软件编程技术已成为计算机学习旳重要内容之一，近几年来，软件编程技术一直在推陈出新，不停有新技术被推出并应用,实在令人眼花缭乱，这些软件编程技术可以分为静态编程技术和动态编程技术两种，静态网页技术一般是指 HTML 和 CSS 技术，采用静态网页技术旳网页只能在网页中展示文字和图片，但它却是所有网页设计旳基础技术，动态网页技术是指让网页可以按照顾客旳需求做出动态响应旳技术,例如 JavaScript,VBScript 及 ASP 等技术都属于动态网页技术旳范围，目前,大部分制作网页旳方式,都是运用网页设计软件,这些软件旳功能相称强大,使用非常以便,不管是那一种网页设计软件,最终都是将所设计旳网页转化为 HTML、CSS、JavaScript/VBScript 和 ASP，因此说,虽然有这些网页设计软件协助我们设计网页,不过一种专业设计者仍需理解 HTML.....等网页设计语言和技术旳使用,这样才能充足让我们理解网页旳真正设计理念。因此毕业设计再次选择这个题目来检查自己旳能力。

我们所用旳软件重要是 Myeclipse6.5，刚运用此软件之前，重要学习 html 基础知识，为学习 java 做铺垫，而重要软件 Myeclipse6.5 是学习旳关键工具，它是设计开发 Web 站点、Web 页和 Web 应用程序旳得力助手。不管是直接编写 HTML 代码还是直接在可视化编辑环境中制作网页，Myeclipse6.5 都会有许多工具协助设计者完毕每项工作，编写出更多完美旳程序。

第二章 开发工具简介

2.1 重要开发工具 Myeclipse6.5 的简介

Myeclipse6.5 是 Genuitec 企业推出的一款非常优秀的 J2EE 集成开发环境，支持代码编写、配置、测试以及除错，它的价格对于个人和企业开发人员来说都是非常有吸引力的。这是 J2EE IDE 市场一种重量级的选手。通过增长 UML 双向建模工具、WYSIWYG 的 JSP/Strutsdesigner、可视化的 Hibernate/ORM工具、Spring 和 Webservices 支持，以及新的 Oracle 数据库开发，MyEclipse 6.5 继续为业界提供全面的产品。

2.2 Myeclipse6.5 特点

1）完整的 CSS 支持

发现可视 CSS 工具的优势，使用这些工具可以轻松地在文献内或文献之间查看、编辑和移动样式，以及查明您的更改将怎样影响设计。借助全新的CSS 布局加速您的工作流，并借助全新的浏览器兼容性检查测试您的设计。

2）集成的编码环境

借助代码折叠、颜色编码、行号及带有注释/取消注释和代码片断的编码工具栏，组织并加速您的编码。应用合用于 HTML 和服务器语言的代码提醒。

3）支持领先的技术

运用对领先 Web 开发技术（包括 HTML、XHTML、CSS、XML、JavaScript、Ajax、PHP、Adobe ColdFusion、ASP、ASP.NET 和 JSP）的支持。

4）轻松的 XML

使用 XSL 或适合于 Ajax 的 Spry 框架，迅速集成 XML 内容。指向 XML 文献或 XML feed URL，Dreamweaver CS3 将显示其内容，这使您可以将合适的字段拖放到您的页面上。

5）FLV 支持

无需任何 Flash 知识，只需五次点击，即可轻松地将 FLV 文献添加到您的 Web 页中。自定义视频环境以匹配您的网站。

6）跨平台知识

挑选您的平台：Myeclipse6.5 可用于基于 Intel 或 PowerPC 的 Macintosh 计算机，也可用于 Windows XP 和 Windows Vista 系统。在您的首选平台中设计，然后跨平台交付愈加可靠、一致和高性能的成果。

2.3 JSP 简介

JSP 全名为 Java Server Pages，中文名叫 java 服务器页面，其主线是一种简化的 Servlet 设计，它是由 Sun Microsystems 企业倡导、许多企业参与一起建立的一种动态网页技术原则。JSP 技术有点类似 ASP 技术，它是在老式的网页 HTML（原则通用标识语言的子集）文献(*.htm,*.html)中插入 Java 程序段(Scriptlet)和 JSP 标识(tag)，从而形成 JSP 文献，后缀名为(*.jsp)。 用 JSP 开发的 Web 应用是跨平台的，既能在 Linux 下运行，也能在其他操作系统上运行。

它实现了 Html 语法中的 java 扩张（以<%,%>形式）。JSP 与 Servlet 同样，是在服务器端执行的。一般返回给客户端的就是一种 HTML 文本，因此客户端只要有浏览器就能浏览。

JSP 技术使用 Java 编程语言编写类 XML 的 tags 和 scriptlets，来封装产生动态网页的处理逻辑。网页还能通过 tags 和 scriptlets 访问存在于服务端的资源的应用逻辑。JSP 将网页逻辑与网页设计的显示分离，支持可重用的基于组件的设计，使基于 Web 的应用程序的开发变得迅速和轻易。JSP(JavaServerPages)是一种动态页面技术，它的重要目的是将表达逻辑从 Servlet 中分离出来。

Java Servlet 是 JSP 的技术基础，并且大型的 Web 应用程序的开发需要 Java Servlet 和 JSP 配合才能完毕。JSP 具有了 Java 技术的简朴易用，完全的面向对象，具有平台无关性且安全可靠，重要面向因特网的所有特点

2.4 Tomcat6.0的简介

Tomcat是Apache软件基金会（Apache Software Foundation）的Jakarta 项目中的一种关键项目，由Apache、Sun 和其他某些企业及个人共同开发而成。由于有了Sun 的参与和支持，最新的Servlet 和JSP 规范总是能在Tomcat 中得到体现，Tomcat 5支持最新的Servlet 2.4 和JSP 2.0 规范。由于Tomcat 技术先进、性能稳定，并且免费，因而深受Java爱好者的爱慕并得到了部分软件开发商的承认，成为目前比较流行的Web应用服务器。目前最新版本是8.0。

Tomcat 服务器是一种免费的开放源代码的Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问顾客不是诸多的场所下被普遍使用，是开发和调试JSP 程序的首选。对于一种初学者来说，可以这样认为，当在一台机器上配置好Apache服务器，可运用它响应HTML（原则通用标识语言下的一种应用）页面的访问祈求。实际上Tomcat 部分是Apache 服务器的扩展，但它是独立运行的，因此当你运行tomcat 时，它实际上作为一种与Apache 独立的进程单独运行的。

诀窍是，当配置对的时，Apache 为HTML页面服务，而Tomcat 实际上运行JSP 页面和Servlet。此外，Tomcat和IIS等Web服务器同样，具有处理HTML页面的功能，此外它还是一种Servlet和JSP容器，独立的Servlet容器是Tomcat的默认模式。不过，Tomcat处理静态HTML的能力不如Apache服务器。

2.5 后台数据库Oracle的简介

ORACLE 数据库系统是美国 ORACLE 企业（甲骨文）提供的以分布式数据库为关键的一组软件产品，是目前最流行的客户/服务器(CLIENT/SERVER)或 B/S 体系构造的数据库之一。例如 SilverStream 就是基于数据库的一种中间件。ORACLE 数据库是目前世界上使用最为广泛的数据库管理系统，作为一种通用的数据库系统，它具有完整的数据管理功能；作为一种关系数据库，它是一种完备关系的产品；作为分布式数据库它实现了分布式处理功能。但它的所有知识，只要在一种机型上学习了 ORACLE 知识，便能在多种类型的机器上使用它。

Oracle 数据库最新版本为 Oracle Database 12c。Oracle 数据库 12*c* 引入了一种新的多承租方架构，使用该架构可轻松布署和管理数据库云。此外，某些创新特性可最大程度地提高资源使用率和灵活性，如 Oracle Multitenant 可迅速整合多种数据库，而 Automatic Data Optimization 和 Heat Map 能以更高的密度压缩数据和对数据分层。这些独一无二的技术进步再加上在可用性、安全性和大数据支持方面的重要增强，使得 Oracle 数据库 12*c* 成为私有云和公有云布署的理想平台

第三章 网站系统分析

3.1 网站系统分析

产品管理系统是亿阳信通企业内部运行的一种测试管理系统，此管理系统可以很便捷的规划员工的工作任务和工作状况及绩效考核。

内容包括：任务工作台、产品管理、验收管理、需求管理、缺陷管理、绩效考核、系统演示等几大模块。

3.1.1 技术可行性分析

在制作学校网站的课程中需要考虑到技术可行性的问题，对制作人员一般都要规定掌握计算机技术，具有一定的数据库技术基础，至少熟悉 JavaSricpt、spring、hibernate、Struts。纯熟运用 Oracle。

3.1.2 操作可行性分析

本作品操作简朴、界面简洁清晰对于那些有一般的计算机知识的人员就可以轻松上手。并兼容与所有网页浏览的软件（如：Internet Explorer）

由此可见，该网站的操作是可行的，有必要开发本网站。

综合以上方面，本网站具有很高的开发可行性，无论从技术上，还是操作上都是可行的。

第四章 详细设计

4.1 文献导入

在 E 盘建立 workspace 文献夹。如图 4-1 所示

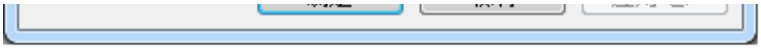


图 4-1 文献夹属性

打开 Myeclipse6.5 →file→Import：进入【导入】窗口，如图 4-2 所示：



图 4-2 Myeclipse 的导入窗口

鼠标单击 Existing Projects into Workspace 再点击底部 Next,进入导入文献选框。如图 4-3 所示：



图 4-3 导入文献选框

双击 Browse 进入系统盘选中要编译的文献夹单击确定。如图 4-4 所示：

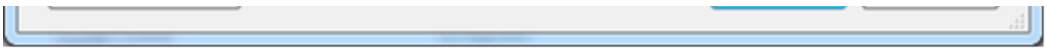


图 4-4 点击确定按钮文献导入结束。

4.2 数据库设计

采用的数据库是 Oracle ，拟建立 My Objects 库文献，包括如下几种表文献。

4.2.1 网站登陆管理数据库 如表 4-5 所示

字段名称	数据类型	备注
ID	自动编号	顾客 ID
epsmsUserName	文本	顾客名
epsmePassword	文本	顾客密码

表 4-5 登陆顾客名和密码

4.2.2 产品管理数据库 如表 4-6 所示

字段名称	数据类型
ID	自动编号
mouldeName	文本
Title	文本
areaId	文本
requiredCompleteTime	时间

status	文本
putForwardUserid	文本

表 4-6 产品管理数据库表

4.2.3 验收管理数据库 如表 4-7 所示

字段名称	数据类型	备注
ID	自动编号	Id
functionName	文本	功能名称
Rank	文本	级别
Processmode	文本	处理方式
Resume	文本	简述

Edition	文本	版本
Otherness	文本	差异描述
Operate	文本	操作列
Relevantmarking	文本	关联标示

表 4-7 验收管理数据库表

4.2.4 需求管理数据库 如表 4-8 所示

字段名称	数据类型	备注
ID	自动编号	ID
areaID	文本	省份
Mouldname	文本	模块
Title	文本	标题
putForwardTime	文本	提出时间
Struts	文本	需求状态
developersUserid	文本	开发人员

表 4-8 需求管理数据库表

4.2.5 缺陷管理数据库 如表 4-9 所示

字段名称	数据类型	备注
id	自动编号	id
areaID	文本	省份
Title	文本	标题
Mouldname	文本	模块
putForwardUserid	文本	提出人
putForwardTime	日期/时间	提出时间
Struts	文本	缺陷状态
developersUserid	文本	处理人员
actualDevelopCompeleteTime	日期/时间	处理时间

表 4-9 缺陷管理数据库表

4.3 JDBC 链接 Oracle 数据库的代码展示

```
<bean id="dataSource"
        class="org.logicalcobwebs.proxool.ProxoolDataSource"
        destroy-method="close">
        <property name="driver">
            <value>oracle.jdbc.driver.OracleDriver</value>
        </property>
        <property name="driverUrl">
            <!--    <value>jdbc:oracle:thin:@127.0.0.1:1521:orcl</value>
-->
```

<!-- <value>jdbc:oracle:thin:@172.30.29.92:1521:eoms</value>

-->

```
    <value>jdbc:oracle:thin:@192.168.0.38:1521:eoms</value>
</property>
<property name="user">
    <value>partner_epsms_test</value>
</property>
<property name="password">
    <value>partner_epsms_test</value>
</property>
<property name="alias">
    <value>platformpool</value>
</property>
<property name="prototypeCount">
    <value>1</value>
</property>
<property name="maximumConnectionCount">
    <value>10</value>
</property>
<property name="minimumConnectionCount">
    <value>1</value>
</property>
<property name="maximumActiveTime">
    <value>100</value>
</property>
<property name="trace">
    <value>true</value>
```

```
    </property>
    <property name="verbose">
        <value>true</value>
    </property>
    <property name="statistics">
        <value>ld</value>
    </property>
</bean>
```

4.4 主页设计

网站主页采用静、动相结合的方式，即静态的主画面和动态的图片相结合，体现了管理页面的简洁明了。

主页构造如图 4-10 所示：

图 4-10　代维管理首页构造图



4.5 产品管理概况

4.5.1 功能

重要对亿阳信通产品管理旳一种基本简介.理解运用过程和运行模式。

4.5.2 构成构造

构成部分：

1、头部是由一种 flash 做成旳为本网页主页起到一种标志性作用。

2、左边栏是由其他链接构成

3、右边栏详细简介本网页主页旳内容

4.5.3 重要的难点代码展示

```
    @SuppressWarnings("unchecked")
       public    ActionForward    showRequireListPage(ActionMapping    mapping,
ActionForm form,
               HttpServletRequest request, HttpServletResponse response)
       throws Exception {

          RequireService                    requireService                    =
(RequireService)this.getBean("requireService");
          Search search = new Search();
          int  firstResult  =  CommonUtils.getFirstResultOfDisplayTag(request,
"requireList");
          String
exportAllFlag=StaticMethod.null2String(request.getParameter("exportAllFlag"));
          String
exportFlag=StaticMethod.null2String(request.getParameter("exportFlag"));
          String[] statusArray = request.getParameterValues("status");
          List<Filter> fls = new ArrayList<Filter>();
          for (int i = 0; statusArray!=null&&i < statusArray.length; i++) {
             String status=statusArray[i];
             fls.add((Filter.equal("status",status)));
          }
          if (fls.size()>0) {
             Filter[] orFileters=new Filter[fls.size()];
```

```
search.addFilterAnd(Filter.or(fls.toArray(orFileters)));
```

```
            }
            if (!"".equals(exportFlag)) {
                if (!"all".equals(exportAllFlag)) {
                    search.setFirstResult(firstResult                                *
CommonConstants.PAGE_SIZE);
                    search.setMaxResults(CommonConstants.PAGE_SIZE);
                }
            }else {
                search.setFirstResult(firstResult * CommonConstants.PAGE_SIZE);
                search.setMaxResults(CommonConstants.PAGE_SIZE);
            }
            search.addSort("provinceId",true);
            search.addFilterEqual("deleted", "0");
            search = CommonUtils.getSqlFromRequestMap(request, search);
            SearchResult<Require>                  searchResult                  =
requireService.searchAndCount(search);
            List<Require> requireList = searchResult.getResult();
            if (!"".equals(exportFlag)) {
                ExcelExportService service=new ExcelExportService();
                service.setResultList(requireList);
                service.Export(mapping, null, request, response);
                return null;
            }
            ITawSystemDictTypeManager                    mgr                    =
(ITawSystemDictTypeManager)getBean("ItawSystemDictTypeManager");
```

```
            ArrayList<TawSystemDictType>              statusList              =
mgr.getDictSonsByDictid("10301");
            ArrayList<String> statusArrayList = new ArrayList<String>();
            if (statusArray!= null) {
                for (int i = 0; i < statusArray.length; i++) {
                    statusArrayList.add(statusArray[i]);
                }
            }
            for (int i = 0; i < statusList.size(); i++) {
                TawSystemDictType   tawSystemDictType   =   (TawSystemDictType)
statusList.get(i);
                if (statusArrayList.contains(tawSystemDictType.getDictId())) {
                    tawSystemDictType.setDictRemark("isTrue");
                }
            }
            request.setAttribute("requireList",requireList);
            request.setAttribute("statusList",statusList);
            request.setAttribute("pagesize", CommonConstants.PAGE_SIZE);
            request.setAttribute("size", searchResult.getTotalCount());



            return mapping.findForward("showRequireListPage");
```

这些代码重要实现数据的导出格式和分页。

4.6 产品管理

4.6.1 功能

重要是实现产品管理，包括数据录入和数据导出。

共四大部分

1、 项目信息管理

2、 产品功能清单管理

3、 文档上传

4、 文档查询



主构造图如图 4-11 所示：

图 4-11 产品管理主页构造图

4.6.2 重要难点代码展示

```
    <script type="text/javascript">
        function deleteInfo(id) {
            if(confirm("确定要删除吗？")){
                Ext.Ajax.request({
                    url:"${app}/",
                    params:{method:"delete",id:id},
                    success:function(res,opt) {
                        Ext.Msg.alert("提醒：
",Ext.util.JSON.decode(res.responseText).infor,function()
{window.location.reload();});
                    },
                    failure:function(res,opt) {
                        Ext.Msg.alert("提醒：
",Ext.util.JSON.decode(res.responseText).infor,function()
{window.location.reload();});
                    }
                });
            }
        }
        var checkflag=false;
        function chooseAll(){
            var objs = document.getElementsByName("checkbox11");
            if(checkflag=="false"){
```

```
            for(var i=0; i<objs.length; i++){
                objs[i].checked="checked";
            }
            checkflag="checked";
        }
        else if(checkflag=="checked"){
            for(var i=0; i<objs.length; i++){
                    objs[i].checked=false;
            }
            checkflag="false";
        }
    }
    function isChecked(){
        /* Ext.Msg.alert('提醒','努力开发中^_^');
        return;
        var objs = document.getElementsByName("checkbox11");
        var flag = false;
        var ids="";
        for(var i=0; i<objs.length; i++){
            if(objs[i].checked==true){
                flag=true;
                if(ids!=""){
                    ids+=",";
                }
                ids+=objs[i].id;
```

```
    }
}
if(flag==false){
    Ext.Msg.alert("提醒","请选择删除项!");
    return;
}
Ext.Msg.confirm("提醒","确认删除?",
    function(btn){
        if(btn=="yes"){
            alert('你点了确定');
        }
    }
); */
var objs = document.getElementsByName("checkbox11");
var ids="";
var flag=false;
for(var i=0; i<objs.length; i++){
    if(objs[i].checked==true){
        flag=true;
        ids+=objs[i].value+",";
    }
}
ids=ids.substring(0,ids.length-1);
if(flag==false){
    Ext.Msg.alert("提醒","请选择删除项!");
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：https://d.book118.com/156150240025010152