**软件系统设计说明书**

兰州家教网站系统设计

团队：Dare To Dream

目录

[第一章 系统概要设计 3](#_Toc516219671)

[**1.1 系统功能模块设计** 3](#_Toc516219672)

[**1.2 系统数据库设计** 7](#_Toc516219673)

[**1.2.1 数据库需求** 8](#_Toc516219674)

[**1.2.2 数据库概念设计** 8](#_Toc516219675)

[第二章 系统详细设计 39](#_Toc516219676)

[**2.1 系统文件结构设计** 39](#_Toc516219677)

[**2.2 模块业务流程设计** 40](#_Toc516219678)

[**2.2.1 教员注册** 40](#_Toc516219679)

[**2.2.2 客户登录** 41](#_Toc516219680)

[**2.2.3 管理员登录** 44](#_Toc516219681)

[**2.2.4 管理员管理** 4](#_Toc516219682)

[**第三章 相关软件技术与开发模式** 6](#_Toc516219683)

[**3.1 开发工具** 6](#_Toc516219684)

[**3.2 开发语言** 7](#_Toc516219685)

[**3.3 开发技术** 9](#_Toc516219686)

[**3.4 开发模式** 10](#_Toc516219687)

[**3.4.1 体系结构** 10](#_Toc516219688)

[**3.4.2 系统软件架构设计** 11](#_Toc516219689)

**第一章 系统概要设计**

概要设计又称总体设计，目的是把数据流图转化为软件结构和数据结构。其中软件结构的设计任务包括对系统进行功能模块划分、建立模块层次结构等；数据结构设计的任务包括描述数据特征、确定数据结构特性、设计数据库等。概要设计为后续系统实现做准备。

**1.1 系统功能模块设计**

根据系统需求分析，家教在线管理平台划分为前台新闻、客户注册与登录、教师客户模块、学生客户模块、家长客户模块、留言消息模块、评论及反馈模块、后台管理员模块等八个功能模块。本系统采用了B/S模式，分为前台和后台，其中系统前台包括新闻、客户注册与登录、教师客户模块、学生客户模块，家长客户模块、留言消息模块、评论及反馈模块；系统后台为管理员管理功能模块。各模块存在功能上的相对独立与结构上的相互关联，例如学生客户功能模块包括查看修改个人信息、搜索教师信息、查看留言消息、发布求聘消息、评论及查看评论回复、浏览新闻等。

(1)访客功能结构图

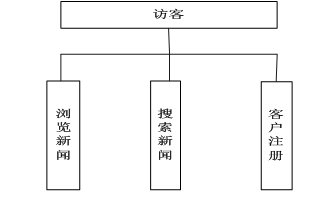


图1-1访客权限功能结构图

访客功能涉及前台新闻、客户注册模块。访客访问本系统网站首页面即可浏览前台新闻、查看新闻详情、按条件搜索新闻信息，再点击“注册”进入注册页面，按要求填写完整、符合规范的注册信息即可成为系统注册客户。

(2)教师客户功能结构图

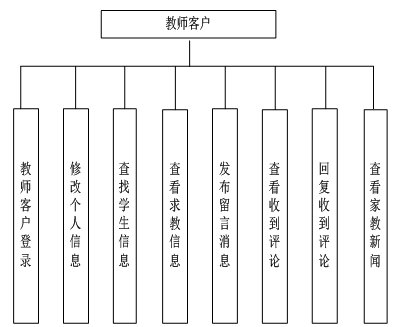


图1-2教师客户权限功能结构图

教师权限功能涉及客户登录模块、教师客户功能模块、学生客户功能模块、留言消息模块、评论及反馈模块、新闻模块。教师客户登录进入权限页面，便可以修改个人信息、查找相关学生信息、查看留言求教信息、发布关于应聘留言消息、查看收到的评论、回复评论、查看家教新闻等。

(3)学生客户功能结构图

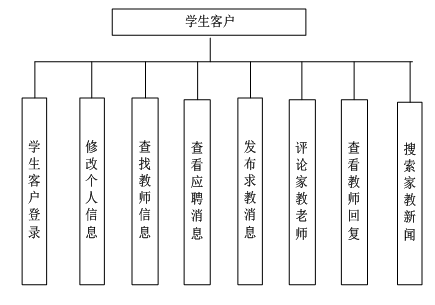


图1-3学生客户权限功能结构图

学生权限功能结构涉及客户登录模块、学生客户功能模块、教师客户功能模块、留言消息模块、评论及反馈模块、新闻模块。学生客户登录进入权限页面，便可以修改个人信息、查找相关教师信息、查看留言应聘信息、发布求教留言消息、评论教师、查看评论回复、查看家教新闻等。

(4)家长客户功能结构图

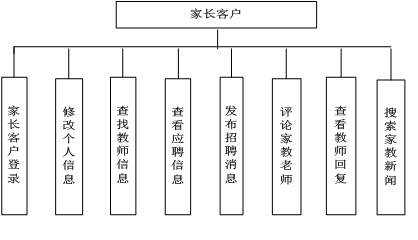


图1-4家长客户权限功能结构图

家长权限功能结构涉及客户登录模块、家长客户功能模块、教师客户功能模块、留言消息模块、评论及反馈模块、新闻模块，家长客户登录进入权限页面，便可以修改个人信息、查找相关教师信息、查看应聘信息、发布招聘消息、评论教师、查看教师回复、搜索家教新闻等。

(5)管理员功能结构图

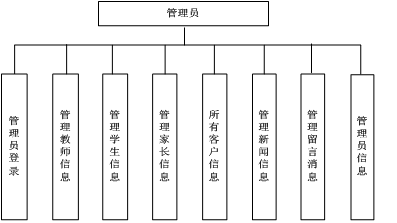


图1-5管理员权限功能结构图

管理员权限功能结构涉及管理员模块、家长客户功能模块、教师客户功能模块、学生客户模块、留言消息模块、新闻模块，管理员登录系统后台，便可以管理不同类别客户以及所有客户信息、管理新闻信息、管理留言消息、查看管理员信息、修改个人信息，超级管理员还能添加或删除管理员。

**1.2 系统数据库设计**

家教在线管理平台系统开发涉及较多数据。系统用户分为前台客户和后台管理员。前台客户包括访客和注册客户。注册客户有学生、教师和家长客户。前台客户浏览、查询站内新闻消息，使用用户名和密码登录到权限页面，使用系统赋予的功能权限，数据流程用到客户信息、留言信息、评论信息。管理员凭借账号和密码登录到系统后台，管理客户信息、留言消息、新闻信息，管理业务流程用到管理员数据、客户数据、留言信息等。系统功能实现需要数据源，需要有稳定安全的后台数据库。数据库在系统中扮演者举足轻重的角色。用户需要的数据存放于数据库中，当用户要这些数据时，可以从数据库中调用。

**1.2.1 数据库需求**

作为系统的内核，数据库需要有良好的设计。在数据库中存储的是稳定性较高的数据。

(1)数据库设计应那满足以下总体性原则：

1)数据表里没有重复冗余的数据。

2)数据表里没有order 1，order 2，order 3等数据。

3)全体数据表的空间占用总量越小越好。

4)使用频率高数据库查询都能以简单高效的方式执行[17]。

(2)选择MYSQL作为数据库。

MYSQL优势较其它数据库如Oracle，Sqlserver等有以下优势：

1)使用C和C++编写，并使用了多种编译器进行测试，保证代码的可移植性。

2)支持AIX、FreeBSD、HP-UX、Linux、Mac OS、Novell Netware、OS/2 Wrap、Solaris、Windows等多种操作系统。

3)为多种编程语言提供了API。这些编程语言包括C、C++、Python、Java、Perl、PHP、Eiffel、Ruby和TCl等。

4)支持多线程，充分利用CPU资源。

5)优化的SQL查询算法，有效地提高查询速度。

6)既能够作为一个单独的应用程序应用在客户端服务器网络环境中，也能够作为一个库而嵌入到其他的软件中提供多语言支持，常见的编码如中文的GB 2312、BIG5，日文的Shift\_JIS等都可以用作数据表名和数据列名。

7)提供TCP/IP、ODBC和JDBC等多种数据库连接途径。

8)提供用于管理、检查、优化数据库操作的管理工具。

MYSQL数据库体积小、速度快、总体拥有成本低，尤其是开放源码，因而本系统选用MYSQL作为后台数据库。

**1.2.2 数据库概念设计**

数据库概念模型独立于现实数据模型，帮助分析表述现实世界中事物和事物间的关系。为了辅助数据库的设计提出实体-联系数据模型。E-R模型是一种语言模型，它将现实世界的含义和相互关联映射到概念模型方面非常有用，因此采用E-R模型分析数据库概念设计[18]。

本系统用户主要有前台客户和系统管理员。前台客户分三类：教师客户、学生客户、家长用户。系统涉及到的实体还包括有 新闻、留言消息、评论。

E-R模型主要包括矩形，菱形，椭圆形和连接线。分析本系统的实体及实体之间的关系，建立实体联系图。

(1)管理员与客户（注册客户）、新闻实体、留言消息实体、评论实体的总体E-R图，如图4-6所示：



图1-6 总体E-R图

由总体E-R图可以清晰得出管理员与新闻、留言消息、用户是管理与被管理的关系，管理员与新闻、客户、留言消息之间相关关系是一对多的。注册客户浏览新闻信息、发布留言信息、使用评论信息，注册客户与新闻、留言、评论信息之间相关关系是一对多的。

(2)客户类别与客户E-R图：

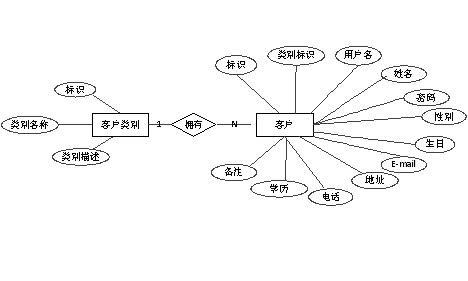


图1-7客户类别与客户E-R图

客户拥有唯一的客户类比属性，一种客户类别包多个客户，客户类别与客户之间是拥有的关系，相关关系是一对多。

(3)管理员与客户E-R图：

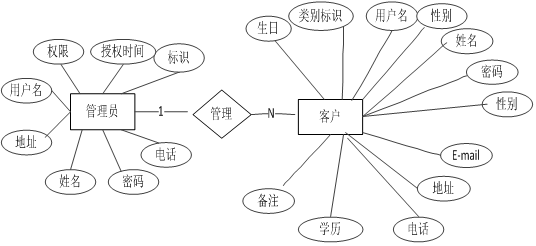
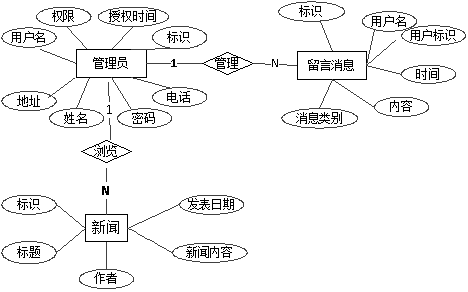


图1-8管理员与客户E-R图

管理员管理多个客户信息，管理员与客户存在一对多的管理与被管理关系。

(4)管理员与新闻信息、留言消息E-R图：



. 图1-9管理员与新闻信息、留言消息E-R图

一名管理员可以管理多条留言消息、管理多条新闻信息，管理员与新闻、留言信息分别存在一对多的管理与被管理关系。

(5)客户与新闻、留言消息、评论实体E-R图：

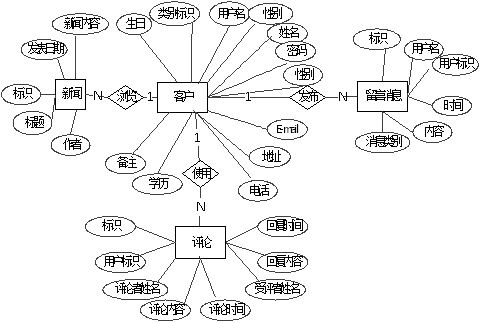


图1-10客户与留言消息、新闻、评论E-R图

客户可以浏览多条新闻信息、可以发布多条留言消息、可以用多条评论，因此客户与新闻信息、留言消息、评论之间存在一对多的相关关系。

# 第二章 系统详细设计

# 本章节依据以上章节的分析，先设计文件结构，再详细设计每个模块实现过程，包括业务流、控制流、功能流程的实现。例如访客注册、注册客户登录、管理员管理等流程，按照先模块流程设计再各模块具体实现来阐述本章节的工作。

**2.1 系统文件结构设计**

本系统实现过程，先按照MVC模式编写代码，代码文件结构为：

项目src文件下存放src文件下存放系统类资源包以及.java文件。其中系统结构中的模型层文件存放于包com.yulin.edu.entity, com.yulin.edu.dao，com.yulin.edu.dao.impl, com.yulin.edu.service, com.yulin.edu.service.impl下,视图层文件存放于WebRoot的资源文件下，控制器文件放于com.yulin.edu.service.action下。图中src下的com.yulin.edu.test包用于存放编码测试文件，com.yulin.edu.util为工具包，存放系统所用工具类文件。

**2.2 模块业务流程设计**

具体设计实现家教在线管理平台访客注册、客户登录、客户使用系统权限功能、管理员登录、管理员管理维护系统信息等业务流程。

**2.2.1 教员注册**

注册页面设计



图2-1注册页面设计

页面要素说明：页面内所有条目都是必填信息，验证用户名不能有重复，密码和确认密码需要填写一致，电子邮箱格式要符合规范。注册角色要求：访客。后续操作：注册成功之后，页面跳转至首页，客户便可用户名和密码登录。完成注册业务流程如下图2-2所示：



图2-2访客注册流程图

注册需要填写注册信息，首先用户名不能与其他已注册用户名重复，然后逐个判断信息是否填写完整正确，只有填写完整正确的注册信息，提交注册才能够注册成功。

功能实现代码：

视图层jsp页面请求

|  |
| --- |
| <form onsubmit=“ return validate()”method=“post”action=“ register.do”>…</form> |

Servlet控制器接受请求并根据请求执行相关动作，调用模型层方法，再把执行结果返回给jsp页面。

接受请求

|  |
| --- |
| String path=uri.substring(uri.lastIndexOf()+1,uri.indexOf());  if(path.equalsIgnoreCase("register")){  try{  action.register(request, response);  response.sendRedirect("../index.jsp");  return ;  }catch(Exception e){  response.sendRedirect("error.jsp");  }  } |

执行动作，即执行ClientAction .java下的register方法

|  |
| --- |
| public void register(HttpServletRequest request, HttpServletResponse response)  throws ServletException, IOException{  request.setCharacterEncoding("utf-8");  response.setCharacterEncoding("utf-8");  String uids=request.getParameter("uid");  int uid=Integer.parseInt(uids);  String uname=request.getParameter("uname");  String name=request.getParameter("name");  String pwd=request.getParameter("pwd");  String gender=request.getParameter("gender");  String birthdays=request.getParameter("birthday");  Date birthday=null;  if(!"".equals(birthdays) && birthdays !=null){  birthday=Date.valueOf(birthdays);  }  String email=request.getParameter("email");  String address=request.getParameter("address");  String phone=request.getParameter("phone");  String edubackground=request.getParameter("edubackground");  String remark=request.getParameter("remark");  Client client=new Client();  client.setAddress(address);  client.setBirthday(birthday);  client.setUid(uid);  client.setUname(uname);  client.setName(name);  client.setPwd(pwd);  client.setGender(gender);  client.setPhone(phone);  client.setEmail(email);  client.setEdubackground(edubackground);  client.setRemark(remark);  ClientService service=new ClientServiceImpl();  try{  service.addClient(client);  }catch(Exception e){  e.printStackTrace();  }  } |

Servlet执行注册动作，调用的模型层的方法有实体Cliebt.java,数据接口ClientDao.java,数据实现ClientDaoImpl.java，业务接口ServiceDao.java,业务实现ServiceDaoImpl.java中的public void addClient(Client client)方法。模型处理的结果发送给控制器，控制器选择合适的视图，生成相应的界面，提供给用户。注册模块，客户注册成功后，页面跳转至系统首页index.jsp,注册客户即可使用用户名和密码登录系统功能页面。

**2.2.2 客户登录**

已注册客户在首页登录处，输入用户名和密码，登录成功后进入功能页面。

角色要求：已注册用户

后续操作：登录验证信息正确，可进入系统功能页面。教师、学生、家长不同类别客户登录进入到不同权限功能页面。登录业务流程如下图2-3所示：



图2-3注册客户登录流程图

注册客户输入用户名和密码，若填写齐全且正确，则客户登录进入前台功能页面，客户分为教师、学生和家长，不同类别客户登录进入不同的权限功能页面。

功能实现代码：

视图层jsp页面请求

|  |
| --- |
| <form name="form1" method="post" action="login.do"></form> |

Servlet控制器接受请求

|  |
| --- |
| ClientAction action=new ClientAction();  String path=uri.substring(uri.lastIndexOf("/")+1,uri.indexOf(".do"));  if(path.equalsIgnoreCase("login")){  action.login(request, response);  return; } |

执行登录动作

|  |
| --- |
| public void login(HttpServletRequest request,HttpServletResponse response)  throws ServletException,IOException{  String uname=request.getParameter("uname");  String pwd=request.getParameter("pwd");  ClientService service=new ClientServiceImpl();  Client client=service.findByLogin(uname, pwd);  if(client!=null){  HttpSession session=request.getSession();  session.setAttribute("client", client);  if(client.getUid()==1){  response.sendRedirect("teacher/tindex.jsp");  }  if(client.getUid()==2){  response.sendRedirect("student/sindex.jsp");  }  if(client.getUid()==3){  response.sendRedirect("parent/pindex.jsp");  }  }else{  response.sendRedirect("./index.jsp");  }  } |

Servlet执行登录动作，调用模型层方法，包括实体Cliebt.java,数据接口ClientDao.java,数据实现ClientDaoImpl.java，业务接口ServiceDao.java,业务实现ServiceDaoImpl.java中的public Client findByLogin(String uname,String pwd)方法。控制器实现登录分流，不同类别客户登录成功进入的权限功能页面不同，教师客户登录进入教师功能页面teacher/tindex.jsp，学生客户登录进入学生功能页面student/sindex.jsp，家长客户登录进入家长功能页面parent/pindex.jsp。

**2.2.3** **客户使用系统权限功能**

客户使用系统功能业务如图2-4所示：



图2-4客户使用系统功能业务流

客户登录进入系统权限页面，行使系统赋予不同类别客户的功能。教师客户进入功能页面之后，可以查看修改个人信息、搜索学生信息、查看求聘消息、发布应聘消息、查看收到的评论以及回复评论；学生客户进入功能页面之后，可以查看修改个人信息、搜索教师信息、查看留言应聘消息、发布求教求聘信息、给教师致信或评论教师等；家长客户进入功能页面之后，可以查看修改个人信息、搜索教师信息、查看留言消息、发布求聘信息、评论教师以及查看评论回复。

具体代码实现：

客户使用系统功能，涉及多个功能子模块。基于系统模型层与控制器的文件层次结构，客户使用系统功能的实现，借助配置文件web.xml中设置的属性参数，将jsp页面的不同动作请求传递给相应的servlet。再由对应servlet判断具体请求，调用不同的action方法和模型层的对应方法。

web.xml中servlet参数配置情况如下：

|  |  |  |
| --- | --- | --- |
| servlet-name | servlet-class | url-pattern |
| ClientActionServlet | com.yulin.edu.action.ClientActionServlet | \*.do |
| AdminActionServlet | com.yulin.edu.action.AdminActionServlet | \*.ad |
| NewsActionServlet | com.yulin.edu.action.NewsActionServlet | \*.news |
| MessageActionServlet | com.yulin.edu.action.MessageActionServlet | \*.message |
| CommentActionServlet | com.yulin.edu.action.CommentActionServlet | \*.comment |

(1)查看修改个人信息

发出动作请求分别是toModify.do与modify.do,请求将被ClientActionServlet.java接受，执行ClientAction.java中的方法public void toModifySelfInfo(HttpServletRequest request, HttpServletResponse response)和public void modifySelfInfo(HttpServletRequest request, HttpServletResponse response)。模型处理的结果发送给控制器ClientActionServlet.java，控制器选择视图selfinfo.jsp,生成客户信息页面，将客户信息展示给客户。

(2)发布留言消息

发出动作请求分别是studentsend.message，parentsend.message，teachersend.message，请求被massageActionServlet.java接受，执行MessageAction.java中的方法public void ClientSendMessageByStyle

(HttpServletRequest request, HttpServletResponse response) 。模型处理的结果发送给控制器MessageAction.java，控制器选择视图sendmessagesuccess.jsp,生成页面，告知客户消息发送成功。

(3)搜索家教新闻

发出动作请求分别是search.news，period.news，请求被NewsActionServlet.java接受，执行NewsAction.java中的方法public void findNewsDim(HttpServletRequest request, HttpServletResponse response)和public void findNewsByPeriod

(HttpServletRequest request, HttpServletResponse response)。 模型处理的结果发送给控制器NewsActionServlet.java，控制器选择视图newslist.jsp生成页面，将搜索到的新闻信息展示给客户。

**2.2****.3 管理员登录**

(1)管理员登录业务流程如图2-5所示：



图2-5管理员登录业务流程图

管理员访问后台登录页面，输入管理员账号和密码，登录成功可进入系统管理后台。

(2)管理员登录代码实现：

发出动作请求是login .ad，请求被AdminActionServlet.java接受，执行AdminAction.java中的方法public void login (HttpServletRequest request, HttpServletResponse response) 。模型处理的结果发送给控制器AdminActionServlet.java，控制器选择视图admin /index.jsp, 即管理员登录进入后台管理系统。

**2.2.4 管理员管理**

(1)管理员登录进入系统管理后台，行使管理权限，后台管理业务流程如下：



图2-6管理员管理功能业务模块

管理员进入系统后台，管理维护家教在线管理平台系统。管理员按权限分为普通管理员和超级管理员。普通管理员与超级管理员进入系统后台拥有不同权限功能，超级管理员可以添加管理员信息、删除管理员信息；普通管理员不能添加删除管理员信息。超级管理员和普通管理员都能够管理不同类别客户信息、管理所有客户信息、管理留言信息、管理新闻信息、维护管理员信息（查看所有管理员和搜索管理员信息）、修改个人信息等。

(2)代码实现：

管理员管理涉及客户、留言信息、新闻、管理员等模块，业务实现过程会发出不同请求，请求为控制器中的servlet接收，执行action中的动作方法，再调用模型层的具体接口及实现方法。

1)管理教师客户信息

视图层发出动作请求是adfindteacherclient.do和findteacherdim.do ,请求被ClientActionServlet.java接受，执行ClientAction.java中的方法public void adminFindTeacherClient(HttpServletRequest request, HttpServletResponse response)和public void adminSearchTeacherClientDim(HttpServletRequest request, HttpServletResponse response)，动作方法再去调用模型层的接口以及实现方法。

模型处理的结果发送给控制器ClientActionServlet.java，控制器选择视图teacherclientlist.jsp,将查询到的教师客户信息列表展示给管理员。

2)管理所有客户信息

视图层发出动作请求是adfindallclient.do 请求被ClientActionServlet.java接收，执行ClientAction.java中的方法public void adminFindClientList(HttpServletRequest request, HttpServletResponse response)，动作方法再去调用模型层的接口以及实现方法。模型处理的结果发送给控制器ClientActionServlet.java，控制器选择视图allclientlist.jsp,将查询到的所有客户信息列表展示给管理员。

3)删除留言信息

视图层发出动作请求是adminremove.message 请求被MessageActionServlet.java接收，执行MessageAction.java中的方法public void adminRemoveMessage(HttpServletRequest request, HttpServletResponse response)，动作方法再去调用模型层的接口以及实现方法。模型处理的结果发送给控制器ClientActionServlet.java，控制器再次执行查找所有留言消息的动作，调用模型层方法，再将处理结果发送给控制器ClientActionServlet.java，选择视图messagelist.jsp,将删除信息后的留言列表展示给管理员。

4)发布新闻信息

视图层发出动作请求是add.news 请求被NewsActionServlet.java接收，执行NewsAction.java中的方法public void addNews(HttpServletRequest request, HttpServletResponse response)，动作方法再去调用模型层的接口以及实现方法。模型处理的结果发送给控制器NewsActionServlet.java，控制器选择视图addnewssuccess.jsp,告知管理员新闻发布成功。

**第三章 相关软件技术与开发模式**

系统开发用到相关软件技术，应该采用较为合适的开发模式，本章节主要介绍系统开发用到的开发工具、语言技术和开发模式。

**3.1 开发工具**

(1)Macromedia Dreamweaver

Macromedia DreamWeaver是建立Web站点和应用程序的专业工具。它将可视布局工具、应用程序开发功能和代码编辑支持组合成一个功能强大的工具，使每一个级别的开发人员都可以利用它便捷地创建界面及站点应用程序[4]。对基于 CSS的设计提供最好的支持到手动编码功能，Dreamweaver在一个集成和高效的环境中为专业人员提供所需的工具开发人员可以将服务器技术(例如CFML、ASP、ASP.NET、JSP、和PHP）与Dreamweaver配合使用，建立连接到数据库 、Web服务和旧时系统的强大Internet应用程序。在Dreamweaver中用户可以创建自己的对象和命令，修改快捷键，编写JavaScript代码、VBScript 代码，用新的行为、属性检查器和站点报告来扩展Dreamweaver的功能。

家教在线管理平台是基于互联网实现的，它面向不同类型用户，需要实现界面友好，考虑到Dreamweaver在网页制作方面的强大功能和使用方便，所以选用Dreamweaver软件来制作网页。Dreamweaver的强大功能体现在：操作方便，有很好的控制力，Dreamweaver是唯一提供Roundtrip HTML视觉化编辑与源码编辑同步的工具，它支持精确定位，可以用图层以拖拉放置的方式来配置版面；制作效率高，Dreamweaver可以用最快的方式将 Fireworks\FreeHand或Photoshop等档案移至网页上 ；方便管理站点，在Dreamweaver中用户可以创建自己的对象和命令，修改快捷键，编写JavaScript代码、VBScript 代码，用新的行为、属性检查器和站点报告来扩展Dreamweaver的功能，使用Dreamweaver可以快速制作网站雏形，设计、更新和重组网页 ；全方位的呈现，利用Dreamweaver设计的网页，可以全方位的呈现在任何平台的热门浏览器上。

(2)Eclipse

Eclipse是一个开放源代码的、基于Java的可扩展开发平台。就其本身而言，它只是一个框架和一组服务，用于通过插件组件构建开发环境。幸运的是，Eclipse 附带了一个标准的插件集，包括Java开发工具（Java Development Kit，JDK）。Eclipse 还包括插件开发环境（Plug-in Development Environment，PDE）。由于Eclipse中的每样东西都是插件，对于给Eclipse提供插件，以及给用户提供一致和统一的集成开发环境而言，所有工具开发人员都具有同等的发挥场所。

这种平等和一致性并不仅限于Java开发工具。尽管Eclipse是使用Java语言开发的，但它的用途并不限于Java语言；例如，支持诸如C/C++和COBOL等编程语言的插件已经可用，或预计将会推出。Eclipse框架还可用作与软件开发无关的其他应用程序类型的基础，比如内容管理系统。

总之，Eclipse是优秀的java集成开发环境，使用它可以很容易地开发java程序。并且Eclipse不是只用作java开发，根据Eclipse的体系结构，通过开发插件，它能扩展到任何语言的开发，甚至成为绘制图片的工具[5]。

**3.2 开发语言**

(1)JSP

JSP(Java Server Pages)是由Sun Microsystems公司倡导、许多公司参与一起建立的一种动态网页技术标准。JSP技术类似ASP技术，它是在传统的网页HTML文件中插入Java程序段和JSP标记(tag)，从而形成JSP文件。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。JSP将网页逻辑与网页设计和显示分离，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容易。Web服务器在遇到访问JSP网页的请求时，首先执行其中的程序段，然后将执行结果连同JSP文件中的HTML代码一起返回给客户。插入的Java程序段可以操作数据库、重新定向网页，能够实现建立动态网页所需要的功能。JSP与Java Servlet一样，是在服务器端执行的，通常返回给客户端的就是一个HTML文本，因此客户端只要有浏览器就能浏览。JSP的1.0规范的最后版本是1999年9月推出的，12月又推出了1.1规范。目前较新的是JSP1.2规范，JSP2.0规范的征求意见稿也已出台JSP页面由HTML代码和嵌入其中的Java代码所组成。服务器在页面被客户端请求以后对这些Java代码进行处理，然后将生成的HTML页面返回给客户端的浏览器。Java Servlet是JSP的技术基础，而且大型的Web应用程序的开发需要Java Servlet和JSP配合才能完成。JSP具备了Java技术的简单易用、完全的面向对象、具有平台无关性且安全可靠、主要面向因特网的所有特点。自JSP推出后，众多大公司都支持JSP技术的服务器，如IBM、Oracle、Bea公司等，所以JSP迅速成为商业应用的服务器端语言。

JSP的特点 ：

1)将内容的生成和显示进行分离[6]。

2)JSP本质就是一个Servlet，提供所有Servlet的功能。

3)在运行的时候，JSP被容器翻译成为一个Servlet，放在tomcat/work/Catalina/localhost目录下。

4)JSP文件被修改之后，可以配置为自动进行重新编译，方便调试。JSP实现了动态页面与静态页面的分离，脱离了硬件平台的束缚以及编译后运行的方式，大大提高了运行效率，继而逐渐成为Internet上的主流开发工具。

JSP作为主流开发技术，可以用一种快捷的方法生成WEB页面，使用JSP技术的WEB页面可以很容易的实现动态内容。JSP能够在任何web或应用程序服务器上运行，将运用程序逻辑或页面显示分离；能够快速的开发和测试；简化开发基于WEB的交互式运用程序的过程[7]。

总而言之，JSP是运行在服务器端的页面，它本身就是一个基于文本的文档，不仅可以包含静态的HTML内容，也可以包含动态的JAVA代码，服务器容器可以将JSP转换成Servlet发布，并接受请求、提供响应。JSP适合作View。

(2)SQL语言

SQL是Struct Query Language的缩写，它是一种标准的关系数据库查询语言。SQL语言分为两类，一类与数据定义有关，称为数据定义语言；另一类与表内的记录有关，称为数据处理语言。常用数据定义语言语句包含Create table(建立新表)、Drop table（删除表）、Alter table（修改表设计结构）。常用数据处理语言语句包含Select 语句（查询）、Insert语句（添加）、Delete语句（删除）、Update语句（更新）。

SQL的主要功能是，它以非过程化形式加快了数据集合的操作。SQL语句表达了用户理想的结果，而且使数据库管理系统负责产生尽可能高效的结果。SQL语句指定了要在行集合上执行操作（如过滤、分组排序）而且数据库系统决定了访问数据的精确方式，以及产生理想结果需要的各种处理步骤的顺序。SQL一个非常有用的方面是“闭包”性质——查询结果以表的形式产生。因此，查询返回的结果集可以插入到另一张表中，或者用作SQL中查询表达式的一部分，如子查询或部分视图定义。

**3.3 开发技术**

(1)Servlet技术

Servlet技术是java技术用于替代CGI编程的一个技术，是java动态web技术的基础，是用Java书写的一种规范，是与平台无关的服务器构件[8]。Servlet程序在服务器端运行，动态的生成WEB页面。与传统的从命令行启动Java应用程序不同，Servlet由Web服务器加载，该WEB服务器必须包含支持Servlet的Java虚拟机，或者与Servlet容器结合使用。Servlet运行在Servlet环境中，Servlet容器负责Servlet和客户的通信以及调用Servlet方法，与其它的动态网页编程技术相比，Servlet有很多特有的优点。

1)高效：Servlet中，每个请求有一个轻量级的Java线程处理（不是重量级的操作系统进程）；如果有N次线程请求，只需要有一份Servlet类代码。

2)方便：Servlet提供了大量的使用工具例程，如自动解析和解码HTML表单数、读取和设置HTTP头、处理Cookie、跟踪会话状态等。

3)功能强大：在Servlet中，许多使用传统CGI程序很难完成的任务都可以轻松地完成。

4)可移植性好：Servlet用Java编写，Servlet API具有完善的标准。因此，为I-Planet Enterprise Server写的无须任何实质上的改动即可移植到Apache、Microsoft IIS或其他的WEB服务器上。

5)节省投资：不仅有许多廉价甚至免费的web服务器可供个人或小规模网站使用，而且对于现有的服务器，如果它不支持Servlet，要加上这部分功能，往往免费。

(2)JDBC技术

JDBC即Java Database Connectivity。JDBC从物理结构上说就是Java语言访问数据库的接口的集合。从本质上说就是调用者和实现者之间的协议。JDBC的设计由SUN公司提出。JDBC的实现由数据库厂商以驱动程序的形式体现 。JDBC API为开发者使用数据库提供了统一的编程接口，包括了一组Java接口和类，使得开发人员能够使用纯Java的方式访问数据库。

**3.4 开发模式**

**3.4.1 体系结构**

不同体系结构的信息系统在开发和运用中有很大的区别，选择合适的体系结构十分重要。主要的信息系统体系结构模式有C/S体系结构、B/S体系结构。

C/S模式体系结构，即客户机/服务器模式。它是由客户机与服务器共同实现的一种协作处理的开发模式，在C/S环境中，将运用功能分成表示层、功能层和数据层，C/S模式有三层和两层之分，C/S模式体系结构多应用于局域网中[9]。

两层C/S结构由两部分构成：前端是客户机，后端是服务器，运行数据库管理系统，提供数据库的查询和管理，表示层与功能层包括显示逻辑和事务处理逻辑，部分放在客户端，而数据层包括数据处理逻辑和数据库，放在服务器端。

两层C/S体系结构如下图2-1所示：



图2-1 两层C/S体系结构

三层结构是将运用功能分成表示层、功能层和数据层三个部分，功能层单独分为中间层部分，这是对两层结构的改进，也是C/S架构中最为常用的。三层C/S架构中，表示层是应用的用户接口，负责处理用户的输入和向客户的输出；功能层是相对于应用的主体，它将具体的业务处理逻辑编入程序中；数据层是数据库管理系统——负责数据库数据读写的管理。如图2-2所示：



图2-2 三层C/S体系结构

B/S体系结构是一种以WEB技术为基础的新型系统平台模式，也就是浏览器、应用服务器、数据库服务器（Browser/WebServer/Database Server简称B/S）的三层体系结构和Internet/Intranet（互联网/企业内部网）[10] 。

B/S 模式是由浏览器、Web服务器和数据库服务器三个层次组成。B/S模式以Web技术为基础，将传统两层C/S模式中的服务器部分又细分为数据库服务器和Web服务器，从而变成三层结构。在B/S模型中，在客户机上运行的应用程序是浏览器。客户机要访问数据库时，向 Web 服务器发出请求，Web服务器通过数据网关与数据库相连，负责与数据库的数据交换。B-S体系结构如图2-3所示：



图2-3 三层B/S体系结构

B/S模式体系结构的系统主要采用四类技术，它们分别是JSP、ASP、PHP、CGI方式。使用B/S体系结构，用户不必安装专门的客户端软件，只需有浏览器如IE、FireFox，就可在任何地方通过通用网路协议与Web服务器交互，继而访问同一数据库数据源。使用B/S结构，客户端有较好的安全性、稳定性和高效性。

系统结构设计本着系统功能全面完善的原则，实现系统的功能结构需求。综合考虑，家教在线管理平台采用B/S体系结构，前台为客户机浏览器，中间件服务器为WEB服务器，后台为数据库服务器。

**3.4.2 系统软件架构设计**

家教在线管理平台系统软件架构采用MVC模式设计。

(1)MVC为一种软件架构思想，是将一个软件的组成部分划分成模型 (model)、视图(view)和控制器(controller)。其中，模型负责业务逻辑(主要包括业务数据的加工处理规则，另外，还有为保证处理所需要的一些基础服务，比如事务、安全、日志等)的处理。视图负责展示模型处理之后的结果，并且提供相应的用户界面或接口，即视图负责表示逻辑。控制器负责协调模型和视图。协调指的是，控制器将视图与模型解藕，这样做的好处是视图或模型发生改变，不会相互影响。

MVC组成结构模块：

|  |  |  |
| --- | --- | --- |
| \*模型  (model) | 负责业务逻辑 | 业务数据的加工规则以及保障业务逻辑能够正常执行所添加的一些基础服务，比如事务、安全、日志、性能等等。 |
| \*视图（view) | 负责表示逻辑 | 业务数据的展现以及用户操作的界面。 |
| \*控制器（controller) | 负责流程控制逻辑 | 视图向控制器发送请求，控制器依据一定的规则，调用对应的模型来处理请求；模型处理的结果发送给控制器，控制器选择合适的视图，生成相应的界面，提供给用户。 |

(2)MVC的处理过程

首先，控制器接受用户的请求，并决定应该调用哪个模型来进行处理，之后模型根据客户请求进行相应的业务逻辑处理，并返回数据；然后控制器调用相应的视图来格式化模型返回的数据，并通过视图呈现给用户[11]。

MVC处理过程如图2-4所示：



图2-4 MVC处理业务过程图

(3)MVC的优点：

1)一个模型可以使用多种不同的视图来展现其处理之后的结果，多个视图能共享一个模型。也就是说，返回的数据是与具体的显示方式无关的。

2)模型开发完之后，可以立即测试。将业务逻辑写在一个java类（即模型）里，可以直接测试该java类，不必启动服务器）。模型是自包含的，与控制器和视图保持相对独立，所以可以方便地改变应用程序的业务数据和业务规则。

3)代码好维护，方便分工协作。控制器提高了应用程序的灵活性。

(4)MVC的两种模式：



图2-5 MVC模式一

JSP Model 1，页面负责调用模型组件来响应客户请求，JSP负责调用模型组件来响应客户请求，并将处理结果返回给用户。JSP既要负责流程控制，还要负责产生用户界面，因此，它要同时充当视图和控制器的功能，未能实现这两个模块之间的独立和分离[12]。

系统中处理业务的逻辑代码嵌套在html的用户界面中，使得表示层和业务层逻辑混乱在一起，这样的页面难以修改与维护，可移植性差，页面的可扩展性及可复用性不好[13]。



图2-6 MVC模式二

JSP Model 2 体系结构，是一种联合使用JSP与Servlet来提供动态内容服务的方法。它吸取了JSP和Servlet两种技术各自的突出优点，用JSP生成表示层的内容，让Servlet完成深层次的处理任务，Servlet充当控制器的角色，负责处理客户请求，创建JSP页面需要的JavaBean对象，并根据客户请求选择合适的JSP页面返回给客户。JSP Model 2体系结构分离了数据展示、数据处理和流程控制，明确了角色定义及软件开发者与网页设计者的分工[14]。

鉴于两种MVC模式各自的优缺点以及本系统开发和后期维护的需要，选择JSP Model 2模式开发本系统软件。