目录

[**系统概要设计** 1](#_Toc515368460)

[**1.1 系统功能模块设计** 1](#_Toc515368461)

[**1.2 系统数据库设计** 4](#_Toc515368462)

[**1.2.1 数据库需求** 5](#_Toc515368463)

[**1.2.2 数据库概念设计** 5](#_Toc515368464)

# 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

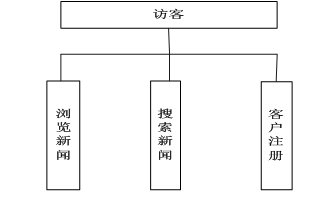


图1-1访客权限功能结构图

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

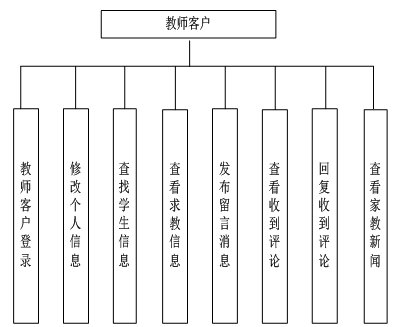


图1-2教师客户权限功能结构图

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

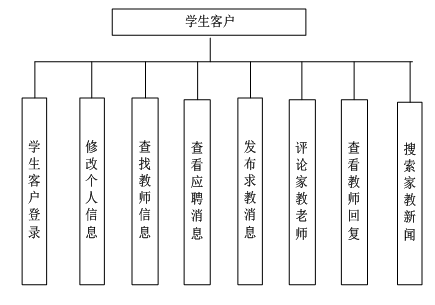


图1-3学生客户权限功能结构图

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

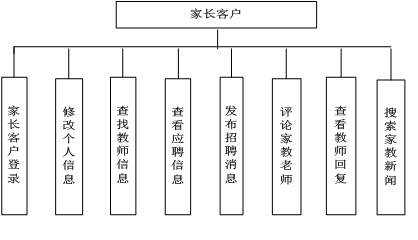


图1-4家长客户权限功能结构图

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

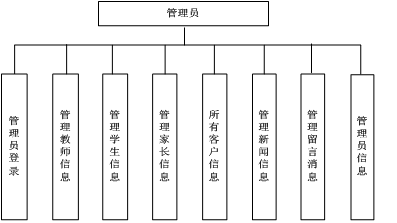


图1-5管理员权限功能结构图

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：

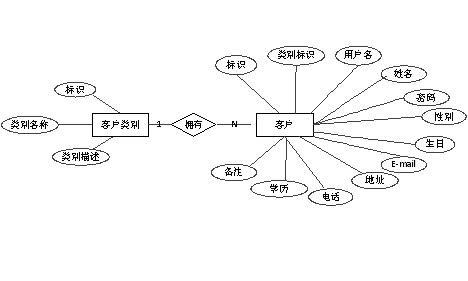


图1-7客户类别与客户E-R图

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：

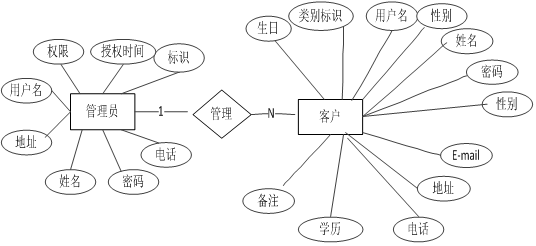
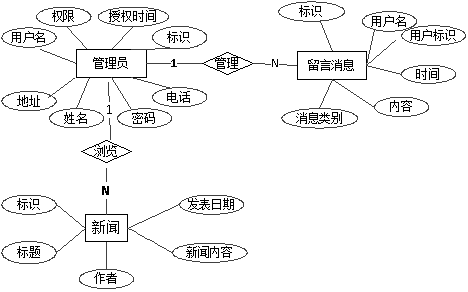


图1-8管理员与客户E-R图

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：

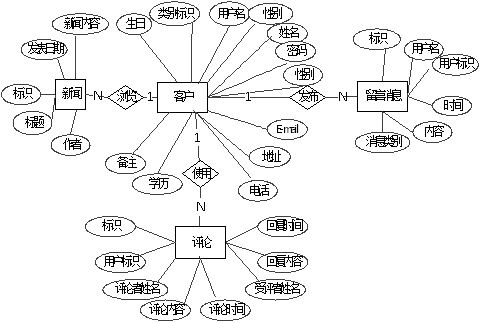


图1-10客户与留言消息、新闻、评论E-R图

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5-1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5-2所示：



图5-2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5-3所示：



图5-3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。

**5.2.3** **客户使用系统权限功能**

客户使用系统功能业务如图5-4所示：



图5-4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5-5所示：



图5-5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5-6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。