# 附件2 网上招聘系统的概要设计说明书

**1.导言**

**1.1 目的**

该文档的目的是描述网上招聘系统项目的概要设计，其主要内容包括：

·系统功能简介；

·系统结构设计；

·系统接口设计；

·数据设计；

·模块设计；

·界面设计。

本文档的预期的读者是：

·开发人员；

·项目管理人员；

·测试人员。

**1.2 范围**

该文档定义了系统的结构和单元接口，但未确定单元的实现方法，这部分内容将在详细设计/实现中确定。

**1.3 编写说明**

UML：Unified Modeling Language（统一建模语言）的缩写，是一个标准的建模语言。

HR：Human Resource（人力资源管理）的缩写。

JSP：Java Server Page（java服务器页面）的缩写，一个脚本化的语言。

MVC:Model-View-Control（模式-视图-控制）的缩写，表示一个三层的结构体系。

EJB:Enterprise Java Bean（企业级JavaBean）的缩写。

**1.4 术语定义**

JSP Model2:Servlet/JSP规范的0.92版本中描述的术语，定义了如何在同一个应用程序中联合使用Servlet和JSP的体系结构。

JavaBean：用Java语言实现的满足一定功能的类。

**1.5 引用标准**

[1]《企业文档格式标准》，\*\*\*\*\*\*\*\*\*\*\*\*软件有限公司

[2]《软件概要设计报告格式标准》，\*\*\*\*\*\*\*\*\*\*\*\*软件有限公司软件工程过程化组织

**1.6 参考资料**

[1]《实战struct》（美）Ted Husted机械工业出版社，

[2]《软件重构》清华大学出版社，

**1.7 版本更新信息**

本文档的更新记录如表B-1所示。

**表B-1 版本更新记录**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 修改编号 | 修改日期 | 修改后版本 | 修改位置 | 修改内容概述 |
| 000  001 | 2022.4.18  2022.4.25 | 0.1  1.0 | 全部  6.5节 | 初始发布版本  修改问卷管理模块的设计 |

**2.系统分析**

本系统可以实现网上在线招聘，应聘者通过互联网投递简历进行网上测评。同时，招聘单位可以汇总简历，游览简历，并通过测评结果选择合格的简历，通知面试，进行面试。方便企业与求职者的交流。系统包括管理端子系统和客户端子系统。

管理端子系统包括题库管理、问卷管理、职位发布、简历管理、面试管理、用户管理等功能。客户端子系统包括查询职位，简历录入，回答问卷，提交简历等功能。图B-1和图B-2为客户端和管理端的组成构图。



**图B-1 客户端子系统图示**   **图B-2 管理端子系统**

**3. 界面设计**

本系统的用户界面按功能分为客户端界面和管理端界面。

删除职位

增加试题

修改试题

删除试题

登录界面

管理首页

问卷管理

职位发布

简历管理

面试管理

题库管理

用户管理

问卷管理

职位发布

简历管理

面试管理

题库管理

用户管理

价值取向问卷

工作取向问卷

职位列表

查询职位

职位详细信息

试题列表

试题详细信息

预览试题

增加职位

修改职位

停止招聘

职位列表

应聘者列表

应聘者申请表

查看试卷答案

回答信息

打印简历

职位列表

应聘者列表

未发通知

等待面试

面试完成

应聘者申请表

查看试卷答案

删除试题

修改试题

增加试题

查询试题

题目列表

题目详细信息

增加答案

删除答案

用户列表

增加用户

删除用户

用户权限设置

用户信息

修改用户信息

**图B-3 管理端的页面流程**

**3.1 管理端界面设计**

管理端主要实现题库管理、问卷管理、职位发布、简历管理、面试管理以及用户管理等功能。主要界面设计如下：

·登录界面：通过输入用户各和密码实现用户登录，并判断用户的权限；



**图B-4 用户登录界面设计**

·管理首页：根据用户的权限，进入首页，并在首页中展示此用户相应可以操作的权限功能；

·管理题库：包括“题目列表”、“题目详细信息”、“增加试题”、“删除试题”、“修改试题”、“增加答案”和“删除答案”等页面；



**图B-5 管理题库界面设计**

·问卷管理：包括“试题列表”、“试题详细信息”、“修改试题”、“增加试题”、“删除试题”和“预览试题”等页面；



**图B-6 管理问卷界面设计**

·发布职位：包括“职位列表”、“职位详细信息”、“查询职位”、“增加职位”、“删除职位”和“修改职位”等页面；



**图B-7 发布职位界面设计**

·管理简历：包括“职位列表”、“应聘者列表”、“应聘者申请表”、“查看试卷答案”、“回复信息”和“打印简历”等页面；



**图B-8 管理简历界面设计**

·面试管理：包括“职位列表”、“应聘者列表”、“应聘者求职申请表”、“查看试卷答案”、“未发通知”、“等待面试”和“面试完成”等页面；



**图B-9 管理面试界面设计**

·用户管理：包括“用户列表”、“用户信息”、“修改用户信息”、“增加用户”和“删除用户”、“用户权限设置”等页面；



**图B-10管理用户界面设计**

**3.2 客户端界面设计**

应聘者网上登录

企业招聘信息

职位列表

职位详细信息

申请职位

填写个人基本信息

填写工作经历

答题

预览简历

填写个人评价

填写家庭成员

填写基本技能

填写教育经历

**图B-11客户端的页面流程**

客户端主要为应聘者提供网上应聘的过程，应聘者通过选择合适的职位，填写个人简历，并通过问卷来进行测评，提交的简历和测评结果一同到服务器端，供管理者挑选合适的简历。

在客户界面，应聘者首先进入企业招聘界面，点击“招贤纳士”按钮进入职位列表界面，包括“职位列表”、“职位详细信息”、“申请职位”、“填写个人基本信息”、“填写工作经历”、“填写教育经历”、“填写基本技能”、“填写家庭成员”、“填写个人评价”、“预览简历”和“答题”等页面。



**图B-12客户端主页面界面设计**



**图B-13应聘者查询职位界面设计**



**图B-14 应聘者查询职位列表界面设计**



**图B-15 应聘者查询职位详细信息界面设计**



**图B-16 应聘者填写简历界面设计**



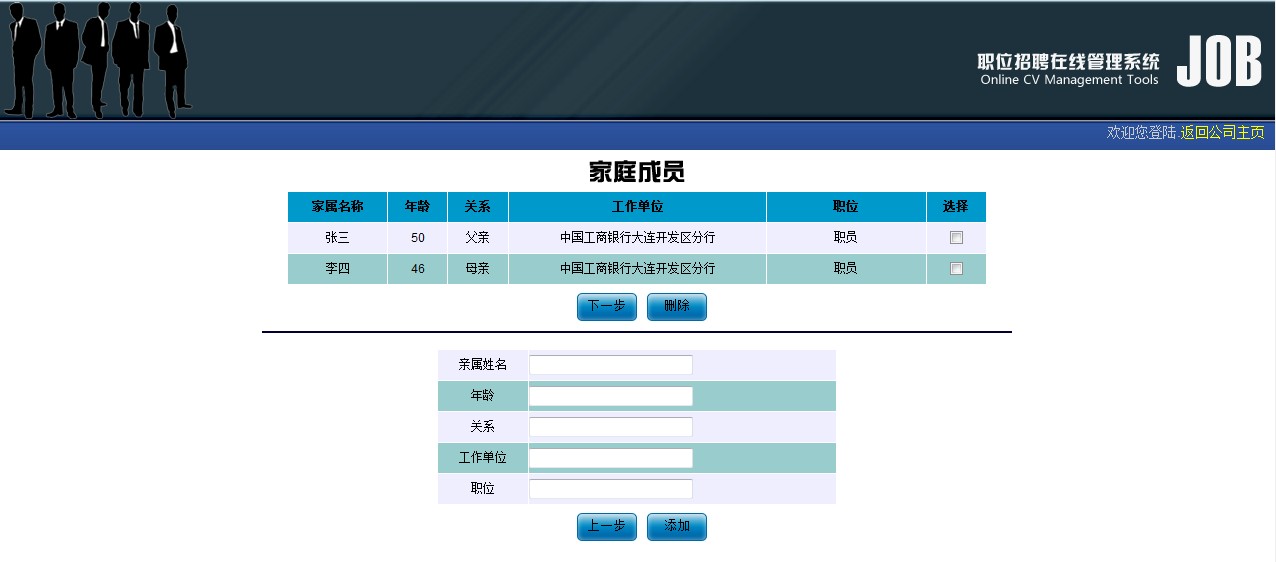
**图B-17 应聘者填写工作经历界面设计**



**图B-18 应聘者填写教育经历界面设计**



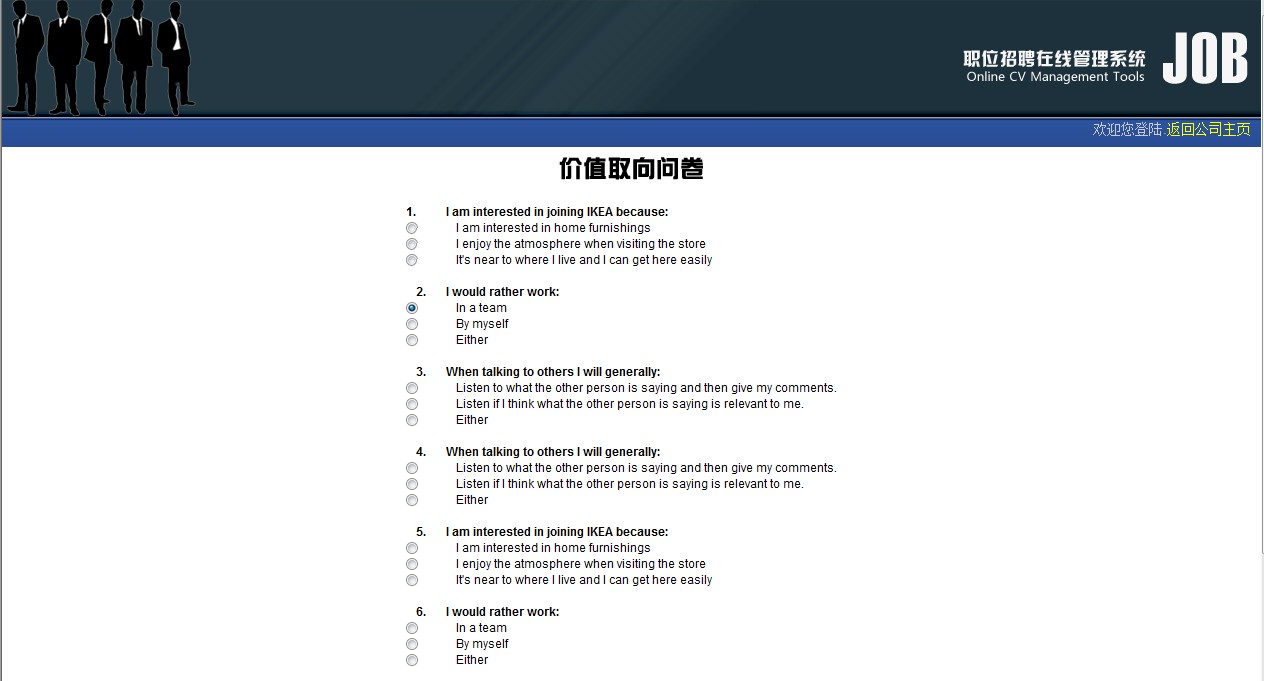
**图B-19应聘者填写基本技能界面设计**

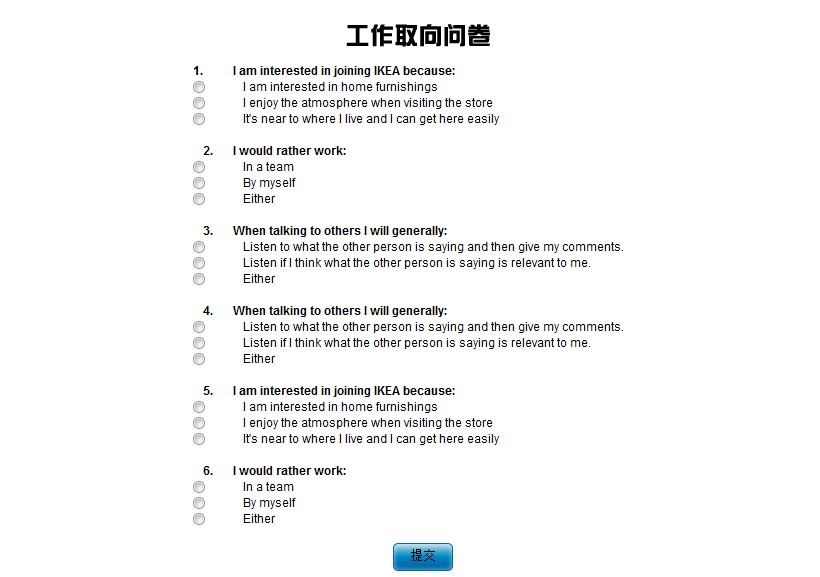


**图B-20 应聘者填写家庭成员界面设计**



**图B-21 应聘者填写自我评价界面设计**





**图B-22 应聘者回答问卷界面设计**

**4. 体系结构**

系统的总体结构设计遵循如下原则。

1）系统应具有良好的适应性：能适应用户对系统的软件环境、管理内容、模式和界面的要求；

2）系统应具有可靠性：采用成熟的技术方法和软件开发平台，以保证系统在以后的实际应用中安全、可靠；

3）系统应具有较好的安全性：应提高安全机制和用户权限限制机制的完善程度，确保数据的受限访问；

4）系统应具有良好的可维护性：系统应易于维护、安装；

5）系统应具有良好的可扩展性：系统应适应未来信息化建设的要求，能方便地进行功能扩展，以建立完善的信息集成管理体系。

本系统采用体系结构，struct是一个基于模型（Model）一视图（View）一控制器（Controller），即MVC模式的应用架构的开源框架。

**4.1 体系结构**

目前软件项目中有很多体系结构，其中struct是比较流行的一种。

**4.1.1 struct体系结构**

对于开发Web应用，要从头设计并开发出一个可靠、稳定的框架不是一件容易的事情。随着Web开发技术的日趋成熟，在Web开发领域出现了一些现成的优秀的框架、开发者可以直接使用它们，struct就是一个很好的框架结构，它是在JSP Model2基础上实现的一个MVC框架，在struct框架在模型由实现业务逻辑的JavaBean或者EJB组件构成，控制器由ActionServlet和Action来实现，视图由一组JSP文件组成，图B-5显示了Struct实现的MVC框架。

其中：

**·视图，**就是一组JSP文件，这些JSP文件没有业务逻辑，也没有模型信息，只有标签，这些标签可以是标准的JSP标签或者是客户化标签，如struct标签库的标签。此外，通常将struct框架中的ActionForm Bean也划为视图模块，ActionForm Bean是一种JavaBean，除了具有一些JavaBean的常规方法外，还包含了一些特殊的方法，用于验证HTML表单数据以及将其属性重新设置为默认值。Struct框架利用ActionForm Bean来进行视图和控制器之间表单数据的传递。Strcut框架将用户输入的表单数据保存在ActionForm Bean中，将它传递给控制器，控制器可以对ActionForm Bean中的数据进行修改，JSP文件使用struct标签读取修改后的ActionForm Bean的信息，然后重新设置HTML表单。

控制器

ActionServlet

视图

JSP

Struct-config.xml

模型

JavaBean

EJB

浏览器

Web

服务器

**图B-23 struct实现的MVC框架**

**·控制器，**控制器由ActionServlet类和Action类实现，ActionServlet类是struct框架中的核心组件，是这个MVC的中央控制器的角色。ActionServlet主要负责接收HTTP请求的信息，根据配置文件struct-config.xml的配置信息，将请求转发给适当的Action对象，如果该Action对象不存在，ActionServlet会先创建这个Action对象.Action类负责调用模型的方法，更新模型的状态，并帮助控制应用程序的流程，对于小型简单的应用，Action类本身也可以完成一些实际的业务逻辑。

**·模型，**模型表示应用程序的状态和业务逻辑，业务逻辑常常由JavaBean或者EJB组件实现。

如果在Web应用开发中套用现成的struct框架，就可以简化每个开发阶段的工作，开发人员可以更加有针对性地分析应用需求，不必重新设计框架，只需在struct框架的基础上，设计MVC各个模块包含的具体组件，在编码过程中，可以充分利用struct提供的各种实用类和标签库，简化编码工作。

Struct框架可以方便迅速地将一个复杂的应用划分成模型、视图和控制器组件，而struct的配置文件struct-config.xml可以灵活地组装这些组件，以简化开发过程。

**4.1.2 系统体系结构**

根据系统分析结果，该系统从结构上应满足：

·基于游览器进行显示以方便用户使用；

·采用MVC的三层体系结构，分化各个功能组件；

·采用JDBC技术与数据库通信以便于数据库的转换；

·采用标签技术完成动态页面的简单逻辑。

本系统的体系结构如图B-6所示，它基本遵循了struct体系的MVC框架规范。

视图(V)层:用户界面(浏览器)

HTML,CSS,DHTML,JavaScript,XML

视图(V)层:服务器端脚本

Connects UI to Business Objects, Java Server Pages,Java Servlets

控制(C)层:分布式组件

JavaBean

模型(M):数据源和持久对象存储

ODBC, JDBC, OLEDB, ADO, XML, LDAP

**图B-24 系统的体系结构**

其中：

**·表示层，**用于与用户进行交互并显示结果。包括所有的JSP，提供用户界面，接受用户输入，还包括相应的ActionFrom Bean，用来存放表单数据，并进行表单数据验证；

**·控制层，**包括所有的Action类，它完成三项任务，一是进行业务逻辑验证，二是调用模型组件，三是决定将合适的视图组件返回给用户；

**·模型，**包括进行逻辑处理的JavaBean等，数据库采用ODBC技术以提供数据库的可移植性。

体系结构的具体拓扑图示如图B-7所示。



**图B-25体系结构拓扑图**

**1）客户层：**用于与企业信息系统的用户进行交互以及显示根据特定业务规则进行计算后的结果。本系统将完全采用基于Web的（B/S架构）客户端，即用户可以直接通过浏览器来访问和使用本系统。

**2）中间层：**这相当于三层标准架构中的Web应用服务层，支持诸如响应客户请求以及查询等功能。并且由中间层进行逻辑处理，再将处理的结果反馈给客户或者发送到数据库中。

**3）服务层：**主要是数据库系统，这里的数据库系统主要是关系数据库系统（RDMS）。

**4.2 系统进行环境**

下面讲述系统运行的网络结构，硬件、软件环境。

**4.2.1 网络结构图**

本系统的网络拓扑图如图B-8所示。



**图B-26 网络拓扑图**

其中的局域网用户机主要是公司内部的人员可以使用的机器，运程用户机主要是指通过互联网登录系统的人员使用的机器，可以是公司内部的人，也可以是应聘者。

**4.2.2 硬件环境**

本系统的硬件环境如下：

客户机：普通PC

· CPU：Intel系列

· 内存：8G以上

· 能够运行IE10.0/Chorme等的机器

· 分辨率：推荐使用1024×768像素

Web服务器

· CPU： Intel系列

· 内存：16G以上

· 硬盘：80TGB以上

· 网卡：KMb/s速度

数据库服务器

· CPU：XEON等

· 内存：32GB以上

·硬盘：80T以上

**4.2.3 软件环境**

本系统的软件环境如下：

· 操作系统：UNIX/Linux/Windows2010或以上版本

· 数据库：SQL Server 2008

· 开发工具包：JDK Version 系列

·Web服务器：Tomcat等

·浏览器：IE10.0以上等

**1） 数据库及操作系统：**对于核心数据库来说，选择一个合适的数据库系统对我们的系统运行是很重要的，选择数据库的关键因素是要考虑预计会有多少人同时访问数据库；正常工作时间的级别；用来访问数据库的应用程序的类型；运行数据库的服务器的硬件和操作系统类型以及管理人员的专业技术水平。目前市场上适用于中小型企业的数据库产品有IBM DB2、Microsoft SQL Server系列，Oracle系列。所有这些产品都基于SQL语言。同时，它们还拥有精度复杂的安全控制以适应不同的商业需要。服务器操作系统使用Windows2010 Server

考虑到价格因素、易用性，我们使用SQL Server2010作为系统后台数据库系统，服务器操作系统采用Windows2010 Server。

**2）Web服务软件：**目前的Web服务器软件有很多种，成熟而且稳定的有Apache、Tomcat和Microsoft的IIS，它们占据着Web服务器市场最大的份额。Tomcat是Sun和Apache合作推出的JSP Server，支持Servlet2.2及JSP1.1等版本。而且Tomcat未来将会取代Jserv，成为Apache主要的Servlet&JSP Engine。Tomcat在设计上是以独立的Server执行，而不像Jserv是附在Apche中，这样就更可以在servlet中，发挥非HttpServlet的能力。Tomcat是Java程序，所以只要有JDK就可以使用，不需要考虑操作系统平台。因此这里选择Tomcat作为Web服务器。

**5. 数据模型**

本系统的数据模型设计内容主要是进行数据库的设计。

**5.1 数据库的概念结构模型设计**

概念设计用来反映现实世界中的实体、属性和它们之间的关系等的原始数据形式，建立数据库的每一幅用户视图。图B-28是系统E-R图。其中系统中的管理用户创建维护题库、管理问卷、发布招聘职位，问卷中的所有题目是由题库中的题目构成，发布的每个职位都对应一套（多个）问卷。应聘者看到招聘职位后，选择合适的职位应聘，提交自己的基本信息、工作经历、教育经历、家庭成员、掌握的基本技能、个人评价等简历情况，并回答问卷。



**图B-27 数据库设计概念模型类图E**

**5.2 数据库的逻辑结构模型设计**



**图B-28 数据库设计的逻辑模型E-R图**

数据库的逻辑设计是将各局部的E-R图进行分解、合并后重新组织起来形成数据库全局逻辑结构，包括所确定的关键字和属性、重新确定的记录结构、所建立的各个数据之间的相互关系。根据本系统需求分析，系统的数据库包括了题库管理、问卷管理、职位发布、简历管理、用户管理以及需要的基本数据字典等部分。

题库管理包括的库表：

·answer——题库中的题目答案；

·question——题库中的题目。

问卷管理包括的库表：

·exam——问卷信息；

·examdetail——问卷包含题目的详细信息；

·jobexam——职位对应的问卷。

职位发布包括的库表：

·joblist——职位的详细列表；

·jobexam——职位对应的问卷。

简历管理包括的库表：

·applyjob——应聘某个职位的信息；

·career——工作经历；

·education——教育经历；

·examresult——问卷回答结果；

·family——家庭关系；

·personalinfo——个人的基本信息；

·skill——基本技能信息。

用户管理包括的库表：

·users——管理端用户的信息。

基本字典包括的库表：

·City——包括的城市名称，例如北京、上海、广州等；

·CurrentSalary——当前的薪水范围，例如1000元以下，1000~2000元等；

·ExpectedSalary——期望的薪水范围，例如1000~2000，2000~3000，……，10000以上等；

·MarriageType——婚姻状况，包括已婚、未婚；

·MoneyType——币种，包括人民币、美元、港元、欧元等；

·Nation——国籍，包括中国、美国、英国、德国等；

·IdentifyType——身份证件类型，包括身份证、军人证等；

·Province——省名，包括黑龙江、吉林、辽宁、河北、河南等；

·HealthType——健康状况类型，包括良好、一般、很好、不好、差等；

·Hukou——户口类型，包括城镇户口、农村户口等；

·CodeJobType——工作类型，包括教师、软件工程师、律师、法官、公务员等。

表B-2是对表Joblist的设计，其他库表的详细设计见数据表设计文件。

**表B-2 Joblist（职位列表）表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 工作经验ID | users | Integer | Y | N |  |
| 申请ID | ApplyID | Integer | Y | N |  |
| 开始日期 | StartDate | Date | N | N |  |
| 结束日期 | EndDate | Date | N | N |  |
| 公司 | Company | varchar | N | N |  |
| 公司类别 | CompanyProperty | varchar | N | N |  |
| 部门 | Depaerment | varchar | N | N |  |
| 职位名称 | Jobtitle | varchar | N | N |  |
| 职位描述 | JobDescription | varchar | N | N |  |

**表B-3申请职位表(ApplyJob)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| ??? | RelId | Ingeter | Y | N |  |
| 申请ID | ApplyID | Ingeter | Y | N |  |
| 状态 | State | varchar | N | Y |  |
| 面试时间 | InterviewTime | Date | N | Y |  |
| 面试官 | InterviewMan | varchar | N | Y |  |
| 面试结果 | InterviewResult | varchar | N | Y |  |

**表B-4职位类型表(JobType)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 职位类型ID | JobTypeID | Integer | Y | N |  |
| 职位类型名称 | JobTypeName | varchar | N | N |  |
| ??? | JobTypeCName | Varchar | N | N |  |

**表B-5答案表(Answer)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 答案ID | AnsID | Integer | Y | N |  |
| 问题ID | QustID | Integer | Y | N |  |
| 内容 | Content | Varchar | N | N |  |
| 评分 | Mark | Double | N | N |  |

**表B-6技能表(Skill)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 技能ID | SkillID | Integer | Y | N |  |
| 申请ID | ApplyId | Integer | N | N |  |
| 技能名称 | SkillName | varchar | N | N |  |
| 技能水平 | SkillLevel | varchaer | N | N |  |
| ？？？ | CertName | varchaer | N | N |  |
| 技能详情 | |SkillDetail | varchaer | N | N |  |

**表B-7职位列表(JobList)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \*字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 职位名称 | JobNAME | varchaer | N | N |  |
| 职位ID | JobID | Integer | N | N |  |
| 职位类型ID | JobTypeID | Integer | N | N |  |
| 发布时间 | PubDate | Date | N | Y |  |
| 结束时间 | EndDate | Date | N | Y |  |
| 职位数量 | JobNum | Integer | N | N |  |
| 描述 | Description | varchar | N | Y |  |
| 需求 | Requirements | varchar | N | Y |  |
| 其它 | Others | varchar | N | Y |  |

**表B-8问题表(Question)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 问题ID | questID | Integer | N | N |  |
| 问题名称 | QuestName | varchar | N | N |  |
| 问题内容 | QuestContent | varchar | N | N |  |
| 问题类型1 | QuestType1 | varchar | N | N |  |
| 问题类型2 | QuestType2 | varchar | N | N |  |
| 其它 | Others | varchar | N | N |  |
| 评注 | Remark | varchar | N | N |  |

**表B-9个人信息表(Personalinfo)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 申请ID | ApplyId | Integer | Y | N |  |
| 职位ID | ApplyJobID | Integer | Y | N |  |
| 姓名 | Name | varchar | N | N |  |
| 英文名 | EngName | varchar | N | N |  |
| 性别 | Gender | Integer | N | N | 1-男，2-女 |
| 生日 | BirthDate | Date | N | N |  |
| 身高 | Height | double | N | Y |  |
| 国籍 | Nation | varchar | N | Y |  |
| 户口 | Hukou | varchar | N | N |  |
| 身份类型 | IdentityType | varchar | N | N | ??? |
| 身份证号 | IdentifyNO | varchar | N | N |  |
| 婚否 | MarriageType | varchar | N | Y |  |
| 工龄 | WorkYear | Integer | N | Y |  |
| 有效时间 | AvailableDate | varchar | N | N |  |
| 目前年薪 | CurrentSalary | varchar | N | Y | ??? |
| 期望年薪 | ExpectedSalay | varchar | N | Y | ??? |
| 住处 | Residency | varchar | N | Y |  |
| 固定电话 | Homephone | varchar | N | Y |  |
| 公司电话 | CompanyPhone | varchar | N | Y |  |
| 移动电话 | moblie | varchar | N | Y |  |
| 电子信箱 | Email | varchar | N | Y |  |

**表B-11考试详情表(Examdetail)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 考试详情ID | ExamDetailId | Integer | Y | N |  |
| 考试ID | ExamID | Integer | Y | N |  |
| 问题ID | questID | Integer | Y | N |  |
| 问题顺序 | QuestOrder | varchar | N | N |  |

**表B-12教育情况表(Education)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 教育情况ID | EducationID | Integer | Y | N |  |
| 申请ID | ApplyID | Integer | Y | N |  |
| 开始日期 | Startdate | Date | N | N |  |
| 结束日期 | EndDate | Date | N | N |  |
| 名字 | name | varchar | N | N |  |

**表B-13家庭情况表(Family)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 姓氏 | FamilyName | varchaer | N | N | ??? |
| 家庭编号 | FamilyID | integer | Y | N |  |
| 申请ID | ApplyID | integer | Y | N |  |
| 学校名称 | SchoolName | varchar | N | N |  |
| 专业 | Major | varchar | N | N |  |
| 年级 | Degree | integer | N | N |  |
| 教育背景 | EduDetail | Varchar | N | N |  |

**表B-14考试表(Exam)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 考试ID | ExamID | Integer | Y | N |  |
| 考试名称 | ExamName | varchar | N | N |  |
| 考试类型 | ExamType | Integer | N | N | 类型介绍 |
| 备注 | Remark | varchar | N | N |  |
| ??? | PassRemark | varchar | N | N |  |
| 发布时间 | IssueDate | Date | N | N |  |
| 职位类型 | JobType | Integer | N | N |  |

**表B-15用户表(Users)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 用户ID | UserID | varchaer | Y | N |  |
| 用户姓名 | UserName | varchar | N | N | 至少是6个英文字母 |
| 用户密码 | UserPwd | varchar | N | N |  |
| 权限 | Permisstion | Integer | N | N | 0-系统管理者  1-面试管理者  2-职位发布者  3-试题管理者  4-普通用户 |

**表B-16** **考试成绩表(Examresult)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| 考试成绩ID | ExamResutId | Integer | Y | N | 至少是6个英文字母 |
| 申请ID | ApplyId | integer | Y | N |  |
| 分数0 | Mark0 | Double | N | N | ??? |
| 答题记录 | AnswerRecord0 | varchar | N | N | ??? |
| 分数1 | Mark1 | double | N | N |  |
| 答题记录 | AnswerRecord1 | varchar | N | N |  |
| 总分数 | Marks | double | N | N | ??? |

**表B-17 职位问题表(JobExam)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段代码 | 字段类型 | 主(外)键 | 是否为空 | 备注(代码字典表) |
| ??? | RelID | Integer | Y | N |  |
| 职位ID | JobID | Integer | Y | N |  |
| 考试ID | ExamID | Integer | Y | N |  |
| 考试类型 | ExamType | varchaer | N | N |  |
| 考试名称 | ExamName | varchaer | N | N |  |

在确定了各个表主键字段的基础上，依据表与表相关字段之间的联系建立了各表之间的关系，如图B-10所示。



**图B-29 数据库设计的物理模型**

**5.3 数据库物理结构模型设计**

信息存储结构的设计在系统的设计中至关重要，要考虑到数据冗余、系统执行效率、信息控制以及维护等方面的要求。信息的管理离不开数据库的支持，我们采用SQL Server 2000数据库管理系统。

数据库的物理设计主要是对数据在内存中的安排，包括对索引区、缓冲区的设计，对使用的外存设备及外存空间的组织，包括索引区、数据块的组织与划分，设置访问数据的方式方法。需在非系统卷（操作系统所在卷以外的其他卷）上安装SQL Server程序及数据库文件。内存是影响Microsoft SQL Server系统性能的一个重要因素，应在Microsoft SQL Server数据库安装后进行内存选项（Memory）设置，最大配置值为2GB。

为了确定SQL Server系统最适宜的内存需求，可以从总的物理内存中减去 Windows 2000 Server需要的内存（12MB）以及其他一些内存需求后综合确定，理想的情况是给SQL Server分配尽可能多的内存，而不产生页面调度。设置服务器的虚拟内存为1GB。对SQL Server属性配置如图B-11所示。

通过ADO对象提供的OLEDB接口与数据库连接。

1）首先创建Connection对象。

Sq1DatabaseName=“WSZP”

Sq1password=“xxxxxxxx”

Sq1username=“sa”

Sq1Loca1Name=“xx.xx.xx.xx”

ConnStr=“provider”=Sqloledb; User ID=“& Sqlusername &”; Password=“& SqlPassword &”;Initial Catalog=“& SqlDatabaseName &”; Data Source=“& SqlLocalName &”;

Set cnn=Server.Createobject (“ADODB.Connection”)

cnn.open Connnstr

2）不直接打开记录集，充分利用连接池。

Set rs=“Server.Createobject”(“ADODB.Recordset”)

rs.oper SQL,cnn

Set rsl=Server.Createobject(“ADODB.Recordset”)

rsl.open SQL,cnn

这种打开连接池的方式可以节省数据库服务器的内存资源。

3）使用记录集后，最早的时间释放数据库资源。

RS.close

set rs=nothing

cnn.close

set cnn=nothing

数据库各库表的脚本代码见数据库脚本文件。

**6. 模块设计**

按照功能分解，本系统分为客户端管理系统和管理端系统。根据页面流的设计，管理端系统又分为用户登录、用户管理、题库管理、问卷管理、职位发布、简历管理、面试管理等7个模块，如图B-30所示。

网上招聘系统

管理端

客户端

登录管理

用户管理

题库管理

问卷管理

职位发布

简历管理

面试管理

**图B-30 模块设计**

各个部分的关系如图B-13所示。

针对不同用户，提供了两类功能登录接口：即应聘者登录接口和企业管理人员登录接口，这些接口都以Web页面的形式提供。通过各自的页面，应聘者和企业管理人员可以从事各自的活动。

以下将分小节介绍各个部分的设计。

**6.1 客户端模块设计**

客户端运行在公用网上，可以显示职位列表，应聘者可以查看所应聘职位的详细信息。当应聘者希望应聘某职位时，可以填写自己的基本信息，包括：姓名、年龄、住址、学历、职业、家庭情况、基本技能、个人评价，这些构成了简历信息，当填写完后可以查看简历。最后应聘者需完成一套问卷试题，试题包括价值取向，工作取向两种类型，题型为选择题，当提交试卷后，简历信息和答题结果存入简历库中，应聘者可以继续应聘其他职位。

在客户端用户可以看到职位名称、职位发布日期、截止日期、职位类型、招聘人数等，当点击职位名称时进入职位详细信息页面。

**6.1.1表示层设计**

根据上述功能介绍，总结出客户端的页面设计如表B-31所列。

管理端页面

客户端页面

用户登录

用户管理

题库管理

问卷管理

职位发布

简历管理

面试管理

职位查询

填写简历

问卷回答

**图B-31 模块之间的关系**

**表B-18 客户端的页面设计**

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 主页面 | index.jsp | 客户端的主页面 |
| 职位列表 | joblist.jsp | 显示招聘的所有职位 |
| 职位详细信息 | jobdetail.jsp | 显示某个职位的详细信息 |
| 个人基本信息 | CVbasicinfo.jsp | 填写个人基本情况页面 |
| 个人工作经历 | career.jsp | 填写个人工作经历页面 |
| 个人教育经历 | education.jsp | 填写个人教育经历页面 |
| 个人基本技能 | skills.jsp | 填写个人基本技能页面 |
| 家庭成员 | relation.jsp | 填写家庭成员页面 |
| 个人评价 | addtion.jsp | 200字以内的个人评价页面 |
| 预览简历 | CVall.jsp | 游览填写完成的简历信息 |
| 答题 | Exam,jsp | 回答试题的页面 |
| 提交 | inputdone.jsp | 提交完成的页面 |

根据界面流的设计可以确定各个界面的访问入口以及界面之间切换关系，流程图如图B-14所示。

ActionForm Bean用于在视图组件和控制组件之间传递HTML表单数据，通常每个HTML表单对应一个ActionForm Bean。此外，ActionForm Bean的validate()方法用于对用户输入的数据进行合法性验证，由于ActionForm Bean工作于视图组件和控制器组件之间，不会访问模型组件，因此，validate()方法通常不涉及对数据的业务逻辑验证，只是完成简单的数据格式和语法检查。表B-4列出了ActionForm以及进行相应的验证。HTML表单中的字段和ActionForm Bean中的属性是一一对应的。

index.jsp

joblist.jsp

jobdetail.jsp

CVbasicinfo.jsp

career.jsp

deucation.jsp

skills.jsp

relation.jsp

addtion.jsp

exam.jsp

inputdone.jsp

CVall.jsp

**图B-32 客户端页面流程图**

**表B-19 客户端的ActionForm**

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 主页面 | index.jsp |  |
| 职位列表 | joblist.jsp | JobForm |
| 职位详细信息 | jobdetail.jsp | JobDetailForm |
| 填写个人基本信息 | CVbasicinfo.jsp | CVbasicinfoFom |
| 填写个人工作经历 | career.jsp | careerForm |
| 填写个人教育经历 | education.jsp | EducationForm |
| 填写个人基本技能 | skills.jsp | SkillsForm |
| 填写家庭成员 | relation.jsp | RelationForm |
| 填写个人评价 | Addtion.jsp | AddtionForm |
| 预览简历 | CVall.jsp | CVallForm |
| 答题 | Exam.jsp | ExamForm |
| 提交完成的页面 | inputdone.jsp |  |

**6.1.2 控制层**

控制层主要是设计Action组件，Action负责单位事件的流程控制，Action映射决定了Action与其他Web组件之间的关联关系。客户端的事件主要包括进入职位列表、浏览职位信息，进入应聘界面，添加应聘者基本信息，添加应聘者工作经历信息，回答试题，提交简历，所有页面的上级返回动作以及工作经历信息、教育经历信息、技能信息、家庭关系信息的增删等动作。表B-5列出了每个Action入口（即调用Action的组件）、传递Action的ActionForm以及出口（即Action将请求转发到目标组件）。

**表B-20 客户端应用的Action映射**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 事件 | Action | 入口 | ActionForm | 出口 |
| 进入职位列表 | EnAction | index.jsp | JobForm | joblist.jsp |
| 浏览职位详细信息 | JobDetailAction | joblist.jsp | JobDetailForm | jobdetail.jsp |
| 进入应聘界面 | ApplyJobAction | jobdetail.jsp |  | CVbasicinfo.jsp |
| 添加应聘者基本信息 | AddbasicinfoAction | CVbasicinfo.jsp | CVbasicinfoForm | career.jsp |
| 工作经历每条信息的增删 | ProcesscareerAction | career.jsp |  | career.jsp |
| 添加应聘者工作经历信息 | AddcareerAction | career.jsp | CareerForm | education.jsp |
| 返回工作经历信息页面的上一步 | BackcareerAction | career.jsp |  | CVbasicinfo.jsp |
| 教育经历每条信息的增删 | ProcesseducationAction | education.jsp |  | education.jsp |
| 添加应聘者教育经历信息 | AddeducationAction | education.jsp | EducationForm | skills.jsp |
| 返回教育经历信息页面的上一步 | BackeducationAction | education.jsp |  | career.jsp |
| 技能每条信息的增删 | ProcessskillsAction | skills.jsp |  | skills.jsp |
| 添加应聘者技能信息 | AddskillsAction | skills.jsp | SkillsForm | relation.jsp |
| 返回技能信息页面的上一步 | BackskillsAction | Skills.jsp |  | education.jsp |
| 家庭关系信息每条信息的增删 | ProcessrelationAction | relation.jsp |  | relation.jsp |
| 添加应聘者家庭关系信息 | AddrelationAction | relation.jsp | RelationForm | addtion.jsp |
| 返回家庭关系信息页面的上一步 | BackrelationAction | relation.jsp |  | skills.jsp |
| 添加应聘者个人评价信息 | AddaddtionAction | addtion.jsp | AddtionForm | exam.jsp |
| 返回自我介绍信息页面的上一步 | BackaddtionAction | addtion.jsp |  | relation.jsp |
| 简历浏览 | CVallAction | addtion.jsp | CVallForm | CVall.jsp |
| 返回简历浏览页面的上一步 | BackCVallAction | CVall.jsp |  | addtion.jsp |
| 提交简历 | SubmitAction | exam.jsp | ExamForm | inputdone.jsp |
| 提交页面返回 | BacksubmitAction | inputdone.jsp |  | joblist.jsp |

**6.1.3 模型层**

在struct框架中，模型组件负责完成业务逻辑，模型组件可以是JavaBean、EJB和实用类。客户端的业务逻辑主要是完成数据库的操作，提交用户的简历信息和问卷回答信息到数据库中。具体的模型组件如表B-6所示。

**表B-21 客户端的模型组件**

|  |  |
| --- | --- |
| 模型组件 | 描述 |
| DBUtil | 数据库的基本操作，为复用组件 |
| CVDataBean | 提交用户的简历信息和试题回答到数据库中 |

**6.2 登录管理模块设计**

登录管理模块负责管理端用户的登录。管理端用户都是通过登录界面进入管理端的，用户输入用户名和密码进入管理界面首页，并根据用户的权限在首页中列出相应的操作功能。

**6.2.1 表示层设计**

根据上述的功能介绍，总结出用户登录的页面如表B-7所示。

**表B-22 登录管理模块的页面设计**

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 登录界面 | login.jsp | 登录的主页面 |
| 主页面 | main.jsp | 管理主页面 |
| 页面中部 | center.jsp | 复用页面：页面中心部分 |
| 页面上端 | top.jsp | 复用页面：页面上面部分 |
| 页面左端 | left.jsp | 复用页面：页面的左面部分 |
| 页面下端 | bottom.jsp | 复用页面：页面的下面部分 |

根据界面流的设计可以确定各个界面的访问入口以及界面之间切换关系，页面的流程图如图B-15所示。



**图B-33 登录管理的页面流程构件图**

**6.2.2 控制层设计**

登录管理的控制层主要是设计用户的登录事件的流程控制（Action）。表B-8列出了每个Action的入口（即调用Action的组件）、传递Action的ActionForm以及出口（即Action将请求转发到目标组件）。

**表B-23 登录管理的控制层设计**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 事件 | Action | 入口 | ActionForm | 出口 |
| 用户 | LoginAction | login.jsp | LoginForm | main.jsp |

**6.2.3 模型层设计**

登录管理模型组件负责完成用户信息的数据库的操作的业务逻辑，建立封装了用户信息的bean，这个bean主要验证用户相关信息是否存在，并判断其权限。模型组件见表B-9

**表B-24登录管理的模型组件**

|  |  |
| --- | --- |
| 模型组件 | 描述 |
| DBUtil  Permission  Userbean | 数据库的基本操作，为复用组件  判断用户权限的类  Bean主要验证用户信息是否存在 |

**6.3 用户管理模块设计**

在用户管理中系统管理员对用户进行增加、删除、修改、查询，进行权限设置。具有不同权限的用户进入不同的主界面，界面左侧栏中的图标有所不同，具体的图标与用户所具有的权限对应。在用户管理中可以增加或删除用户，编辑用户名，修改用户权限等。

**6.3.1 表示层设计**

根据上述功能介绍，总结出用户管理功能的页面如表B-25。

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能描述 |
| 用户界面  用户详细信息  修改用户信息  添加用户 | main.jsp  userlist.jsp  userdetail.jsp  adduser.jsp | 用户管理主界面  用户详细信息的界面  完成特定用户信息的维护。即修改删除等  增加用户信息 |

用户管理模块各个表示页面之间的关系如图B-34所示。



**图B-34 用户管理页面之间的关系构件图**

**6.3.2 控制层设计**

用户管理的控制层主要负责进入用户信息界面、时入浏览用户信息界面、增加用户信息、修改用户信息、删除用户信息等事件的流程控制，即设计用户管理的Action。表B-11列出了用户管理的每个Action的入口（即调用Action的组件）、传递Action的ActionForm以及出口（即Action将请求转发到目标组件）。

**6.3.3 模型层设计**

用户管理的模型层主要是完成用户信息的浏览、增加用户信息、用户信息维护等业务逻辑，并完成相应的数据库操作。模型组件如表B-27所列。

**表B-26 用户管理的控制层设计**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 事件 | Action | 入口 | ActionForm | 出口 |
| 进入用户信息界面 | UserlistAction | main.jsp | UserListForm | userlist.jsp |
| 进入浏览用户信息界面 | UserDetailAction | userlist.jsp | UserDetailForm | userdetail.jsp |
| 修改用户信息界面 | ModifyUserAction | userlist.jsp | ModifyUserForm | userdetail.jsp |
| 提交修改用户信息界面 | SubmitModifyAction | userdetail.jsp | ModifyUserForm | userlist.jsp |
| 增加用户信息 | AddUserAction | userlist.jsp | AddUserForm | userdetail.jsp |
| 提交增加用户信息 | SubmitAddAction | userdetail.jsp | AdduserForm | userlist.jsp |
| 删除用户信息 | DelUserAction | userlist.jsp | DelUserForm | userlist.jsp |

**表B-27 用户管理的模型组件**

|  |  |
| --- | --- |
| 模型组件 | 描述 |
| DBUtil | 数据库的基本操作，为复用组件 |
| Userbean | 主要完成管理用户信息 |

**6.4 题库管理模块设计**

题库管理主要包括：对知识库的维护和管理，即对题库的题目进行的增加、删除、修改的功能及提供知识库中题目的详细信息。

**6.4.1 表示层设计**

根据上述的功能介绍，总结出题库管理功能的页面设计如表B-28所示。

**表B-28 题库管理的页面设计**

|  |  |  |
| --- | --- | --- |
| 界面 | JSP | 功能 |
| 试题信息首页 | kninlist.jsp | 题库管理主页面 |
| 试题信息列表 | knlist.jsp | 显示知识试题列表 |
| 试题详细信息 | kndetail.jsp | 知识试题的详细信息 |
| 增加试题 | addkn.jsp | 增加试题的页面 |
| 增加试题（购物车） | addkn\_cart.jsp | 增加试题的购物车 |
| 增加答案 | addanswer.jsp | 增加试题答案的页面 |
| 增加答案（购物车） | addanswer\_cart.jsp | 增加试题答案的购物车 |

注：题库管理模块中也用到前面提到的公共页面，例如left.jsp,top.jsp.bottom.jsp。

各个界面流的基本流程关系如图B-17所示。



**图B-35 题库管理的页面构件图**

**6.4.2 控制层设计**

题库管理的控制层主要负责进入试题信息界面，进入试题详细信息界面，进入增加试题信息。删除试题信息界面，进入增加答案界面等事件的流程控制。表B-14列出了题库管理控制层每个Action的入口（即调用Action的组件）、传递Action的ActionForm以及出口（即Action将请求转发到目标组件）。

**6.4.3 业务逻辑层设计**

题库管理业务逻辑层设计主要包括建立封装了知识信息的bean，建立封装了答案信息的bean以及封装答案的购物车，它们完成将试题和相应的答案存放于数据库的操作，同时也提供了数据维护的操作等逻辑。模型组件见表B-29。

**表B-29 题库管理的控制层设计**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 事件 | Action | 入口 | ActionForm | 出口 |
| 进入试题信息列表 | QuestionSearchAction | kninlist.jsp | QuestionSearchForm | knlist.jsp |
| 进入试题详细信息 | QuestionDetailAction | knlist.jsp | QuestionDetailForm | kndetail.jsp |
| 增加试题 | AddQuestionAction | knlist.jsp | AddQuestionForm | addkn.jsp |
| 删除试题 | DelQuestionAction | knlist.jsp | DelQuestionForm | knlist.jsp |
| 增加答案 | SubmitAnswerAction | addanswer.jsp | SubmitAnswerForm | kninlist.jsp |
| 删除答案 | DelAnswerAction | kndetail.jsp | DelAnswerForm | kninlist.jsp |
| 增加答案（购物车） | SubmitAnswer\_CartAction | addanswer\_cart.jsp | SubmitAnswer\_CartForm | kn\_cart.jsp |
| 删除答案（购物车） | DelAnswer\_CartAction | addanswer\_cart.jsp | DelAnswer\_CartForm | kn\_cart.jsp |

**表B-30 题库管理的模型组件**

|  |  |
| --- | --- |
| 模型组件 | 描述 |
| DBUtil | 数据库的基本操作，为复用组件 |
| Questionbean | 封装了知识信息的bean |
| Answerbean | 封装了答案信息的bean |
| KnCart | 封装答案的购物车 |

**6.5 问卷管理模块设计**

问卷管理是管理员根据知识题库中的试题生成各种类别的问卷，并且可以对问卷的试题进行增加、删除、修改的功能。

**6.5.1 表示层设计**

根据上述的功能介绍，总结出问卷管理功能的页面如表B-16所示。

各个界面基本流程如图B-36所示。



**图B-36 问卷管理的页面构件图**

**表B-31 问卷管理的页面设计**

|  |  |  |
| --- | --- | --- |
| 页面 | JSP | 功能描述 |
| 问卷列表 | examlist.jsp | 显示所有问卷列表，可以删除问卷 |
| 问卷详细信息 | examdetail.jsp | 显示每个问卷的具体信息，可以上下移动试题，可以删除试题 |
| 增加问卷 | addexam\_cart.jsp | 增加问卷 |
| 增加试题 | questionadd.jsp | 增加问卷中的试题 |
| 预览问卷 | examview.jsp | 浏览问卷 |

注：问卷管理页面中也使用到了公共的页面，例如left.jsp,top.jsp,bottom.jsp等。

**6.5.2 控制层设计**

问卷管理控制层主要负责进入问卷信息界面、进入问卷详细信息界面、进入增加问卷信息、删除问卷信息界面、进入增加试题界面、浏览问卷等事件的流程控制。表B-17列出了问卷管理控制层每个Action的入口（即调用Action的组件）、传递Action的ActionForm以及出口（即Action将请求转发到目标组件）。

**6.5.3 模型层设计**

问卷管理的业务逻辑主要是完成问卷维护，并完成相应数据库的操作。所以问卷管理的模型层主要是建立封装了问卷信息bean的以及封装了数据库操作的组件。模型组件见表B-32。

**表B -32 问卷管理控制层设计**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 事件 | Action | 入口 | ActionForm | 出口 |
| 进入问卷列表 | ExamSearchAction | examlist.jsp | ExamSearchForm | examlist.jsp |
| 进入问卷详细信息 | ExamDetailAction | examlist.jsp | ExamDetailForm | examdetail.jsp |
| 增加问卷 | AddExamAction | examlist.jsp | AddExamForm | examlist.jsp |
| 删除问卷 | DelExamAction | examlist.jsp | DelExamForm | examlist.jsp |
| 增加试题 | QuestionSearchAction | examdetail.jsp | 无 | questionadd.jsp |
| 提交试题 | SubmitQuestionToExamAction | questionadd.jsp | SubmitQuestionToExamForm | examdetail.jsp |
| 删除试题 | DelQuestionExamAction | examdetail.jsp | DelQuestionExamForm | examdetail.jsp |
| 预览问卷 | ViewExamAction | examdetail.jsp | ViewExamForm | viewexam.jsp |
| 上下移动试题 | OrderQuestAction | examdetail.jsp | OrderQuesForm | examdetail.jsp |

**表B-33 问卷管理的模型组件**

|  |  |
| --- | --- |
| 模型组件 | 描述 |
| DBUtil | 数据库的基本要操作，为复用组件 |
| Exambean | 封装了问卷信息的bean |
| ExamCart类 | 封装问卷的购物车 |

**6.6 职位发布模块设计**

职位发布模块主要完成招聘职位的编制和发布，在职位列表中显示职位名称，职位发布日期，截止日期，职位类型，招聘人数等，按职位发布日期排序。当点击职位名称时进入职位详细信息页面，职位详细页面显示职位名称，职位描述，职位要求，招聘人数，点击应聘该职位进入填写基本信息页面。

**6.6.1 表示层设计**

根据上述的功能介绍，总结出职位发布功能的页面如表B-34所列。

**表B-34 职位发布的页面设计**

|  |  |  |
| --- | --- | --- |
| 页面 | JSP | 功能描述 |
| 职位列表 | joblist.jsp | 所有职位列表 |
| 职位详细信息 | jobdetail.jsp | 每个职位详细信息，提供删除职位功能 |
| 修改职位信息（更新） | modifyjob.jsp | 修改职位信息 |
| 修改职位信息（提交） | modifyjobsubmit.jsp | 提交修改后的职位信息 |
| 增加职位（更新） | addjob.jsp | 提交增加职位、更新信息 |
| 查询职位 | jobsearchdetail.jsp | 查询职位 |
| 增加价值、工作取向试卷 | selectexam.jsp | 增加价值、工作取向试卷 |
| 修改增加价值取向试卷 | modifyselectvalueexam.jsp | 修改增加价值取向试卷 |
| 修改增加价值取向试卷（提交） | modifyvaluesubmit.jsp | 修改增加价值取向试卷（提交） |
| 修改增加工作取向试卷 | modifyselectjobexam.jsp | 修改增加工作取向试卷 |
| 修改、删除增加工作取向试卷（提交） | modifyjobsubmit.jsp | 修改、删除增加工作取向试卷（提交） |

注：职位发布模块的页面也使用了公共页面，例如left.jsp,top.jsp,bottom.jsp等。

各个页面流程如图B-37所示。



**图B-37 职位发布模块的页面设计构件图**

**6.6.2 控制层设计**

职位发布的控制层主要负责进入职位信息界面、进入职位详细信息界面、进入增加职位信息、删除试卷信息界面、进入增加试卷界面等事件的流程控制。表B-20列出了职位发布控制层每个Action的入口（即调用Action的组件）、传递Action的ActionForm以及出口（即Action将请求转发到目标组件）。

**表B-35 职位发布控制层设计**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 事件 | Action | 入口 | ActionForm | 出口 |
| 进入职位列表 | JobListAction | Index.jsp(main.jsp) |  | joplist.jsp |
| 进入职位详细信息 | JobDetailAction | joplist.jsp | JobDetailForm | jobsearchdetail.jsp |
| 修改职位信息 | ModifyJobAction | jobdetail.jsp | ModifyJobForm | modifyjob.jsp |
| 修改职位信息（更新） | UpdateModifyJobAction | modifyjob.jsp | UpdateModifyJobForm | modifyjobsubmit.jsp |
| 修改职位信息（提交） | ModifyJobSubmitAction | Modifyjobsubmit.jsp | ModifyJobSubmitForm | joplist.jsp |
| 增加职位 | AddJobPreAction | joplist.jsp |  | addjob.jsp |
| 增加职位（更新） | UpdateAction | addjob.jsp | UpdateForm | addjobl.jsp |
| 增加职位（提交） | AddJobAction | addjobl.jsp | AddJobForm | joplist.jsp |
| 删除职位 | DelJobAction | jobdetail.jsp | DelJobForm | joplist.jsp |
| 查询职位 | JobSearchAction | joplist.jsp | JobSearchForm | jobsearchdetail.jsp |
| 增加价值取向试卷 | AddJobValueExamPreAction | aAddjobl.jsp | AddJobValueExamPreForm | selectexam.jsp |
| 增加价值取向试卷（提交） | AddJobValueAction | Selectexam.jsp | AddJobValueExamForm | addjobl.jsp |
| 从购物车中删除价值取向试卷 | UpdateJobCartAction | addjobl.jsp | UpdateJobCartForm | addjobl.jsp |
| 增加工作取向试卷 | AddJobExamPreAction | addjobl.jsp | AddJobExamPreForm | selectjobexam.jsp |
| 增加工作取向试卷（提交） | AddJobExamAction | selectjobexam.jsp | AddJobExamForm | addjobl.jsp |
| 从购物车中删除工作取向试卷 | DelJobExamAction | addjobl.jsp | DelJobExamForm | addjobl.jsp |
| 修改增加价值取向试卷 | ModifyJobValueExamPreAction | modifyjobsubmit.jsp | ModifyJobValueExamPreForm | modifyselectvalueexam.jsp |
| 修改增加价值取向试卷（提交） | ModifyJobValueExamAction | modifyselectvalueexam.jsp | ModifyJobValueExamForm | modifyvaluesubmit.jsp |
| 修改增加工作取向试卷 | ModifyJobExamPreAction | modifyjobsubmit.jsp | ModifyJobExamPreForm | modifyselectjobexam.jsp |
| 修改增加工作取向试卷（提交） | ModifyJobExamAction | modifyselectexam.jsp | ModifyJobExamForm | modifyvaluesubmit.jsp |
| 删除修改价值取向试卷 | ModifyDelValueExamAction | modifyvaluesubmit.jsp | ModifyDelValueExamForm | modifyselectvalueexam.jsp |
| 删除修改工作取向试卷 | ModifyDelJobExamAction | modifyvaluesubmit.jsp | ModifyDelJobExamForm | modifyvaluesubmit.jsp |

**6.6.3 模型层设计**

职位发布模型的业务逻辑主要是完成职位维护，对职位所附的试卷进行操作，并完成相应数据库的操作。所以职位发布的模型层有如下模型：

·建立封装了数据库操作的组件；

·建立封装了职位基本信息的组件；

·建立封装了职位试卷信息的组件；

·建立封装了职位试卷购物车信息的组件；

·建立封装了试卷基本信息的组件（同问卷管理模块的ExamBean）。

模型组件见表B-36。

**表B-36 职位发布的模型组件**

|  |  |
| --- | --- |
| 模型组件 | 描述 |
| DBUtil | 数据库的基本操作。为复用组件 |
| Jobbean | 封装了职位住处的bean |
| JobExambean | 封装职位问卷的信息 |
| JobCartbean | 封装职位问卷的购物车 |
| ExamBean | 封装问卷信息 |

**6.7 简历管理模块设计**

简历管理主要是人力资源部门的人员对应聘者的简历进行检查、审查和整理的工作。

**6.7.1 表示层设计**

根据上述的功能介绍，简历管理功能的页面如表B-37所示。

**表B-37 简历管理的页面设计**

|  |  |  |
| --- | --- | --- |
| 页面 | JSP | 功能描述 |
| 职位列表 | jobResumelistp.jsp | 职位列表，每个职位下有相应的简历 |
| 人员简历列表 | personinfo.jsp | 人员简历列表 |
| 简历详细信息界面 | personinfoDetail.jsp | 简历详细信息界面 |
| 简历回复界面 | reply.jsp | 简历回复界面 |

注：简历管理模块的页面也使用了公共页面，例如left.jsp,top.jsp,bottom.jsp等。

各个页面流程如图B-38所示。

main.jsp

jobResumelist.jsp

Personinfo.jsp

personinfoDetail.jsp

personinfoDetail.jsp

Reply.jsp

**图B-38 简历管理的页面流程图**

**6.7.2 控制层设计**

简历管理模块的控制层主要负责进入简历主信息界面、进入简历详细信息界面、进入简历回复界面等事件的流程控制。表B-38列出了简历管理控制层每个Action的入口（即Action调用的组件）、传递Action的ActionForm以及出口（即Action将请求转发到目标组件）。

**表B-38 简历管理的控制层设计**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 事件 | Action | 入口 | ActionForm | 出口 |
| 进入简历主界面 | ResumeAction | Main.jsp |  | Jobresumelist.jsp |
| 进入简历界面列表 | ResumelistAction | Jobresumelist.jsp | joblistForm | jobresumelist.jsp |
| 进入简历详细信息界面 | PersoninfoAction | jobresumelist.jap |  | Personinfo.jsp |
| 浏览个人简历信息 | PersoninfoDetailAction | Personinfo.jsp |  | personinfoDetail.jsp |
| 进入简历回复界面 | InReplyAction | Personinfodetail.jsp |  | Reply.jsp |
| 提交简历回复界面 | ReplyAction | Reply.jsp | replyForm | PersoninfoDetail.jsp |

**6.7.3 模型层设计**

简历管理模型的业务逻辑主要是完成简历浏览，对简历的回复等操作，并完成相应数据库的操作。模型组件见表B-39。

**表B-39 简历模块的模型组件**

|  |  |
| --- | --- |
| 模型组件 | 描述 |
| DBUtil | 数据库的基本操作。为复用的组件 |
| Jobbean | 建立封装了职位信息的bean |
| Replybean | 封装了回复信息的bean |

**6.8 面试管理模块设计**

面试管理主要是人力资源部门的人员完成对简历的检查、审查和整理的工作，通知面试，进行面试，记录面试结果。

**6.8.1 表示层设计**

根据上述的功能介绍，总结出面试管理功能的页面如表B-40所列。

**表B-40 面试管理的页面设计**

|  |  |  |
| --- | --- | --- |
| 页面 | JSP | 功能描述 |
| 面试主界面 | interview.jsp | 显示简历模块的主页面功能 |
| 面试者的目前状态 | interviewState.jsp | 显示应聘者的各种状态：面试过、等待面试、保留简历 |
| 面试者信息界面 | CVdetail.jsp | 显示面试者简历页面 |
| 面试者面试结果界面 | interviewResult.jsp | 填写面试者面试结果的界面 |

注：面试管理模块的页面也使用了公共页面，例如left.jsp,top.jsp,bottom.jsp等。

各个页面流程如图B-39所示。

main.jsp

interview.jsp

interviewState.jsp

CVdetail.jsp

CVdetail.jsp

interviewResult.jsp

**图B-39 面试管理的页面流程图**

**6.8.2 控制层设计**

面试管理的控制层主要负责进入面试信息主界面，进入应聘者详细信息界面、进入显示接受面试的应聘者的不同状态信息界面、进入面试界面等事件的流程控制。表B-41列出了每个Action的入口（即调用Action的组件）、传递Action的ActionForm以及出口（即Action将请求转发到目标组件）。

**表B-41 面试管理控制层设计**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 事件 | Action | 入口 | ActionForm | 出口 |
| 进入面试信息主界面 | InInterviewAction | Main.jsp |  | interview.jsp |
| 进入面试管理界面 | InterviewAction | interview.jsp |  | interviewState.jsp |
| 进入等待面试人员界面 | WaitAction | interviewState.jsp | InterviewStateForm | interviewState.jsp |
| 进入面试过的人员的页面 | InterviewedAction | interviewState.jsp | interviewStateForm | interviewState.jsp |
| 浏览面试人员的简历 | ViewResumeAction | interviewState.jsp | CVdetailForm | CVdetail.jsp |
| 进入面试过程界面 | InterviewStateAction | CVdetail.jsp |  | interviewResult.jsp |
| 面试结束 | InterviewResultAction | interviewResult.jsp | interviewResultForm | CVdetail.jsp |

**6.8.3 模型层设计**

面试管理模型的业务逻辑主要是完成简历浏览，对面试者状态整理、面试结果的处理等操作，并完成相应数据库的操作，模型组件见表B-42。

**表B-42 面试管理的模块组件**

|  |  |
| --- | --- |
| 模型组件 | 描述 |
| DBUtil | 数据库的基本操作。为复用组件 |
| Jobbean | 建立封装了职位信息的bean |
| Interviewbean | 封装了面试信息的bean |