<省人力资源市场数据采集系统 >

概要设计说明书

版本 <1.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 16/12/2015 | <0.1> | 功能设计 | BITSS |
| 19/12/2015 | <0.1> | 设计系统架构 | BITSS |
| 22/12/2015 | <1.0> | 设计总体类图 | BITSS |
| 23/12/2015 | <1.0> | 完善类设计 | BITSS |

目录

1. 简介 5

1.1 目的 5

1.2 范围 5

1.3 定义、首字母缩写词和缩略语 5

1.4 参考资料 5

1.5 概述 5

2. 整体说明 5

2.1 产品总体效果 5

2.2 产品功能 5

2.3 用户特征 5

2.4 约束 6

3. 具体需求 6

3.1 功能 6

3.1.1 用户登录模块 6

3.1.2 人力资源市场用户模块 8

3.1.3 市级用户模块 9

3.1.4 省级用户模块 14

3.1.5 系统管理员模块 19

3.2 可靠性 20

3.2.1 准确性 21

3.2.2 系统用户安全认证 21

3.2.3 用户权限控制 21

3.2.4 日志安全 21

3.2.5 数据备份与恢复 21

3.3 性能 21

3.3.1 时间特性 21

3.3.2 适用性 21

3.4 扩展性要求 21

3.5 运行环境规定 22

3.6 设计约束 23

3.6.1 人力资源市场用户信息 23

3.6.2 人力资源用户上报报表格式及数据要求 23

3.7 联机用户文档和帮助系统需求 23

3.8 购买的构件 23

3.9 接口 23

3.9.1 用户界面 24

3.9.2 硬件接口 24

3.9.3 软件接口 24

3.9.4 通信接口 24

3.10 许可需求 24

3.11 法律、版权及其他声明 24

3.12 适用的标准 24

4. 支持信息 24

软件需求规约

# 简介

## 目的

本文档是省人力资源市场数据采集系统项目的概要设计说明书。本文档明确了系统的整体结构及业务流程，目的包括：对系统进行总体功能设计，其中包括对每个模块处理流程的分析；对数据库的设计，对程序接口的定义及出错处理等。

预期读者为项目开发人员及维护人员。

## 范围

本文档适用于省人力资源市场数据采集系统项目，用于开发和测试的指导。

## 定义、首字母缩写词和缩略语

SRS:软件需求规格说明书。

HLD:软件概要设计说明书

LLD:软件详细设计说明书

监测点用户：人力资源市场用户。

市用户：市就业管理部门。

省用户：省就业管理部门。

假删除：信息仍在，但处于失效状态。

## 参考资料

1. 《Java编程思想》
2. 《软件工程》
3. 《设计模式》
4. 《省人力资源市场数据采集系统需求规格说明书》

# 项目背景描述

## 运行环境

|  |  |  |
| --- | --- | --- |
| 编号 | **名称** | **运行环境** |
| **1** | 应用服务器 | Web服务器 |
| **2** | Web服务器 | Tomcat 6.0.26 |
| **3** | 数据库 | MySQL |
| **6** | 客户端 | IE8及以上浏览器，Firefox, Google chrome |

## 设计和实现上的约束

|  |  |  |
| --- | --- | --- |
| 限制因素 | 限制说明 | 备注 |
| 必须采用的技术、工具、编程语言、数据库等 | B/S混合结构，数据库采用mysql数据库。其他无特殊限制 |  |
| 不能使用的技术、工具、编程语言、数据库等 | 无特殊限制 |  |
| 企业策略、政策法规、业界标准 | 必须遵守中国人民共和国的相关法律法规， |  |
| 硬件限制 | 无特殊限制 |  |
| 性能限制 | 无特殊限制 |  |

## 假定和依赖

* 假设因素

表格 2‑3假设因素

|  |  |  |
| --- | --- | --- |
| 编号 | 假设 | 备注 |
| 1 | 客户端操作系统IE8.0以上或firefox3.0以上版本 |  |
| 2 | 对现有业务、功能描述与实际情况基本相符合，对需求的变更不影响系统框架大的调整。 |  |

* 依赖因素

表格 2‑4依赖因素

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 依赖 | 依赖说明 | 备注 |
| 1 | Mysql | 业务数据存储在mysql数据库中 |  |
| 2 | Tomcat | 系统Web发布通过tomcat实现 |  |

## 需求概述

本系统由四种角色：监测点用户，市级用户，省级用户，系统管理员。

省管理部门创建服务机构账号；就业服务机构通过账号登录系统，补充各级服务机构信息，经市管理部门备案后，市管理部门进行审核，全部通过后，报送省级管理部门；市级管理部门审核、汇总各区县公共及经营性人力资源服务机构报送的数据，上报省厅；省厅对数据进行汇总。

系统管理员可以管理省用户。

|  |  |  |
| --- | --- | --- |
| 功能类别 | 功能名称 | 一般过程描述 |
| 人力资源市场用户 | 用户信息 | 修改人力资源市场用户基本信息 |
| 数据填报 | 填报人力资源市场用户监测数据 |
| 数据查询 | 查询人力资源市场用户以往调查期数据状态 |
| 市  区县 | 管理人力资源市场用户 | 创建监测点账号和基本信息 |
| 上报备案 | 对当前调查期选择需录入数据的人力资源市场用户，上报备案 |
| 用户填报情况 | 查看审核人力资源市场用户上报的数据 |
| 报表管理 | 上报本期数据，查询往期数据 |
| 数据退回 | 退回上报数据 |
| 数据汇总 | 查询汇总表 |
| 数据导出 | 按报送期导出人力资源市场用户信息、报表等数据 |
| 数据查询 | 对本市已创建用户进行条件查询 |
| 发布通知 | 发布、删除通知信息 |
| 数据分析 | 分析全市人力资源市场用户采集数据 |
| 省 | 企业备案 | 查看各市已备案监测点信息 |
| 企业查询 | 按需要对备案监测点进行查询 |
| 报表管理 | 审核地市上报的数据并汇总上报？给谁 |
| 数据退回 | 退回上报数据 |
| 数据汇总 | 查询汇总表 |
| 数据导出 | 按报送期导出监测点信息、报表等数据 |
| 数据查询 | 对全省已创建用户进行条件查询 |
| 取样分析 | 分析全省备案人力资源市场用户各行业的占比情况 |
| 图表分析 | 用图表方式分析人力资源市场用户数据变动情况 |
| 固定报表 | 输出六类固定格式报表到word文件中 |
| 发布通知 | 发布、删除通知信息 |
| 系统管理 | 设置上报时限、管理用户、监控系统运行情况 |
| 系统管理员 | 管理省用户 | 修改省用户信息，新增或删除省用户 |
| 管理市用户 | 修改市用户信息，新增或删除市用户 |
| 系统监控 | 监控系统运行情况 |

# 总体设计

## 处理流程

下面对主要的模块中主要处理流程进行描述。

### 用户登录模块



### 监测点用户信息备案模块



### 报表数据上报模块







### 系统管理模块



# 系统的技术架构说明

系统由5层结构组成，分别为表示层、控制层、业务逻辑层、持久化层和数据库层。如下图所示。其中，表示层由JSP 页面组成，提供程序与用户之间交互的界面；业务逻辑层用于处理程序中的各种业务逻辑；控制层主要由Servlet 进行控制；持久化层由Hibernate 框架组成，负责应用程序与关系型数据库之间的操作；数据库层为应用程序所使用的数据库，本系统采用MySQL 数据库。

表示层（JSP）

控制层（Servlet）

业务逻辑层

实体层

数据持久层（Hibernate）

数据库

# 系统的静态架构说明

## 类列表

本系统分为四个层级（Action（控制器类）、Service（业务领域类）、Dao（持久类）、Entity（实体层））来写，四个表中都包含类名称、类属性和类方法；除此外还包括一个工具包（Tool）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **层级** | **类名称** | **类说明** | **类属性** | **类方法** |
| **Action** | **1.DataServlet** | 数据报表控制类 | -role:String;表示用户身份 | -saveData():int;//保存填写内容  -report\_c():int;//上报给市级  -report\_p():int;//上报给省级  -pass\_p():int;//省通过审批  -back():int;//数据退回  -export():int;//数据导出  -summary\_c():int;//市数据汇总  -summary\_p():int;//省数据汇总  -selectData():int;//获取数据  -search():int;//搜索  -comparative\_analysis():int;//对比分析  -trend\_analysis():int;//趋势分析  -checkRight():int;验证用户权限 |
| **2.UserServlet** | 用户控制类 | -role:String;表示用户身份 | +login():int;//登录  +change\_info():int;//修改信息  +record():int;//备案  +change\_pwd():int;//修改密码  +createUser():int;//创建用户  +deleteUser():int;//删除用户  +getUser():int;//查询用户信息 |
| **3.AnnouncementServlet** | 通知控制类 | -role:String;表示用户身份 | +release();//发布通知  +delete();//删除通知  +sellectAll();//浏览通知 |
| **4.InvestServlet** | 调查期控制类 | -role:String;表示用户身份 | +create();新增上报时限  +chage();修改上报时限 |
| **5.SystemMonitorServlet** | 系统监控控制类 | -role:String;表示用户身份 | +getSystemState();查看系统状态 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **层级** | **类名称** | **类说明** | **类属性** | **类方法** |
| **Service** | **1.ManageData** | 处理报表数据的业务逻辑 |  | +saveData(Data data):int;//保存已填写数据  +report\_c(String username):int;//监测点上报给市级  +report\_p(String username):int;//市级将监测点数据上报给省级  +pass\_p(String username):int;//省级通过监测点数据审批  +back(String username):int;//数据退回  -summary\_c(String cityname):Data;//市数据汇总  -summary\_p(String provname):Data;//省数据汇总  -selectDataByUser(String userid,String begintime,String endtime):Data;//获取监测点数据  -selectDataByCity(String cityname,String begintime,String endtime):Data;//获取市数据  -selectDataByProvince(String provname,String begintime,String endtime);//获取省数据  -exportByMon(String time);//监测点用户导出本期数据  -exportByCity(String time);//市用户导出数据  -exportByProv(String time);//省用户导出数据  -search\_p(String key);//省模糊搜索，对用户ID，名称，表名，调查期匹配  -search\_c(String key);//市模糊搜索，对用户ID，名称，表名，调查期匹配  -search\_m(String key);//监测点模糊搜索，对用户ID，名称，表名，调查期匹配  -comparative\_analysis(String username1,String username2,String time);//两个监测点对比分析  -trend\_analysis(String username,String begintime,String endtime);//趋势分析  -checkRight(String username);验证用户权限 |
| **2．ManageUser** | 处理用户及用户信息的业务逻辑 |  | +add\_user(User u):int;//增加新用户  +remove\_user(String id):int;//删除用户  +update\_Moniuser(Moni\_User u):int;//更新监测点用户信息  +update\_Cityuser(City\_User u):int;//更新市用户信息  +update\_Provuser(Prov\_User u):int;//更新省用户信息  +update\_Systemuser(System\_User u):int;//更新系统管理员用户信息  +update\_pwd(String id):int;//修改密码  +get\_user\_by\_id(String id):User;//通过用户id查询用户信息  +get\_user\_by\_name(String name):User;//通过用户名查询用户信息  +check\_by\_id(String id):int;//通过用户id查询用户权限 |
| **3.ManageAnnouncement** | 处理通知的业务逻辑 |  | +release(Announcement ann ):int;//发布通知  +delete(String ann\_id);//删除通知  +sellectAll():List<Announcement>;//浏览通知 |
| **4．Manage\_Investperiod** | 处理调查期及上报时限的业务逻辑 |  | +create(String begintime,String endtime):int;新增上报时限  +change(String begintime,String endtime);修改上报时限 |
| **5.** **ManageSystemState** | 处理系统监控的业务逻辑 |  | +getSystemState():SystemState;查看系统状态 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **层级** | **类名称** | **类说明** | **类属性** | **类方法** |
| **DAO** | **1．DataDao** | 与报表有关的数据库操作类 |  | +saveData(Data data):int;//保存已填写数据  +report\_c(String username):int;//监测点上报给市级  +report\_p(String username):int;//市级将监测点数据上报给省级  +pass\_p(String username):int;//省级通过监测点数据审批  +back(String username):int;//数据退回  -summary\_c(String cityname):Data;//市数据汇总  -summary\_p(String provname):Data;//省数据汇总  -selectDataByUser(String userid,String begintime,String endtime):List<Data>;//获取监测点数据  -selectDataByCity(String cityname,String begintime,String endtime):List<Data>;//获取市数据  -selectDataByProvince(String provname,String begintime,String endtime):List<Data>;//获取省数据  -exportByMon(String time):int;//监测点用户导出本期数据  -exportByCity(String time):int;//市用户导出数据  -exportByProv(String time):int;//省用户导出数据  -search\_p(String key):List<Data>;//省模糊搜索，对用户ID，名称，表名，调查期匹配  -search\_c(String key):List<Data>;//市模糊搜索，对用户ID，名称，表名，调查期匹配  -search\_m(String key):List<Data>;//监测点模糊搜索，对用户ID，名称，表名，调查期匹配  -comparative\_analysis(String username1,String username2,String time):List<Data>;//两个监测点对比分析;  -trend\_analysis(String username,String begintime,String endtime):List<Data>;//趋势分析  -checkRight(String username):int;验证用户权限 |
| **2.UserDao** | 与用户有关的数据库操作 |  | +add\_user(User u):int;//增加新用户  +remove\_user(String id):int;//删除用户  +update\_Moniuser(Moni\_User u):int;//更新监测点用户信息  +update\_Cityuser(City\_User u):int;//更新市用户信息  +update\_Provuser(Prov\_User u):int;//更新省用户信息  +update\_Systemuser(System\_User u):int;//更新系统管理员用户信息  +update\_pwd(String id):int;//修改密码  +get\_user\_by\_id(String id):User;//通过用户id查询用户信息  +get\_user\_by\_name(String name):User;//通过用户名查询用户信息  +check\_by\_id(String id):int;//通过用户id查询用户权限 |
| **3.AnnouncementDao** | 与通知有关的数据库操作 |  | +release(Announcement ann ):int;//发布通知  +delete(String ann\_id);//删除通知  +sellectAll():List<Announcement>;//浏览通知 |
| **4.** **Invest\_periodDao** | 与调查期有关的数据库操作 |  | +create(String begintime,String endtime):int;新增上报时限  +change(String begintime,String endtime);修改上报时限 |
| **5.SystemMonitorDao** | 与系统监控有关的数据库操作 |  | +getSystemState():SystemState;查看系统状态 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **层级** | **类名称** | **类说明** | **类属性** | **类方法** |
| **Entity** | **1.Data\_Items** | 数据项 | -demand\_num:int 需求人数  -demand\_rate:int 需求比重  -demand\_mom:int 需求人数同比  -demand\_yoy:int 需求人数环比  -apply\_num:int 求职人数  -apply\_mom:double 求职人数同比  -apply\_yoy:double 求职人数环比  -app\_dem\_rate:double 求人倍率  -vac\_dem\_rate:double 岗位空缺与求职人比率 | 每个类属性包含一个get()方法和set()方法 |
| **2.Total** | 供求总体人数表 | Total:DataItems | 每个类属性包含一个get()方法和set()方法  init():void初始化数据和项目 |
| **3.Domain** | 按产业分组的需求人数表 | allData:ArrayList<Map<String, Data\_Items>> | 同上 |
| **4.Profession** | 按行业分组的需求人数表 | allData:ArrayList<Map<String, Data\_Items>> | 同上 |
| **5.Character** | 按用人单位性质分组的需求人数表 | allData:ArrayList<Map<String, Data\_Items >> | 同上 |
| **6.Occupation** | 按职业分组的供求人数表 | allData:ArrayList<Map<String, Data\_Items>> | 同上 |
| **7.DemandOverApply** | 需求大于求职缺口最大的前十个职业表 | allData:ArrayList<Map<String, Data\_Items>> | 同上 |
| **8.ApplyOverDemand** | 需求小于求职缺口最大的前十个职业表 | allData:ArrayList<Map<String, Data\_Items>> | 同上 |
| **9.ApplySort** | 按求职人员类别分组的求职人数表 | allData:ArrayList<Map<String, Data\_Items>> | 同上 |
| **10.GenderSort** | 按性别分组的供求人数表 | allData:ArrayList<Map<String, Data\_Items>> | 同上 |
| **11.AgeSort** | 按年龄分组的工区人数表 | allData:ArrayList<Map<String, Data\_Items>> | 同上 |
| **12.DegreeSort** | 按文化程度分组的供求人数表 | allData:ArrayList<Map<String, Data\_Items>> | 同上 |
| **13.SkillSort** | 按技术等级分组的供求人数表 | allData:ArrayList<Map<String, Data\_Items>> | 同上 |
| **14.Data** | 报表类 | Total供求总体人数表  Domain按产业分组的需求人数表  Profession按行业分组的需求人数表  Character按用人单位性质分组的需求人数表  Occupation按职业分组的供求人数表  DemandOverApply需求大于求职缺口最大的前十个职业表  ApplyOverDemand需求小于求职缺口最大的前十个职业表  ApplySort按求职人员类别分组的求职人数表  GenderSort按性别分组的供求人数表  AgeSort按年龄分组的工区人数表  DegreeSort按文化程度分组的供求人数表  SkillSort按技术等级分组的供求人数表 | 同上 |
| **15.User** | 用户父类 | -user\_id:String;//用户编号  -user\_name:String;//用户姓名  -password:String;//用户密码  -checked:int;//是否验证（1表示已验证，0表示未验证）  -role:String;//角色（1表示省级用户，2表示市级用户，3表示监测点用户）  -valid:int;//是否可用（1表示可用，0表示被禁用）  -belong:String;//所属市  -unit:String;//标注用户来源（省、市、监测点）  -is\_deleted;//是否被删除（0表示被删除，1表示未被删除） | 每个类属性包含一个get()方法和set()方法 |
| **16.Moni\_User** | 监测点用户继承自User类 | -login\_id:String;//监测点登陆人id  -site\_code:String;//监测点编码  -site\_name:String;//监测点名称  -site\_region:String;//所属地区  -profession:String;//行业  -site\_address:String;//监测点地址  -liable\_person:String;//责任人  -mobile\_number:String;//手机号码  -phone\_number:String;//电话号码  -fax\_number:String;//传真号码  -valid:int;//监测点信息是否上报备案（1表示已上报备案，0表示未备案） | 同上 |
| **17.City\_User** | 市级用户继承自User类 | -city\_name:String;//城市名  -city\_code:String;//城市编码 | 同上 |
| **18.Prov\_User** | 省级用户继承自User类 | -province\_name:String;//省名  -provice\_code:String;//省编码 | 同上 |
| **19.System\_User** | 系统管理员继承自User类 | -system\_name:String;//管理员姓名  -system\_code:String;//管理员编码 | 同上 |
| **20.Announcement** | 通知通告类 | -id:String  -title:String  -context:String  -time:String  -publisher:String | 同上 |
| **21.Invest\_period** | 调查期类 | -month:String  -begindate:String  -enddate:String | 同上 |
| **22.SystemState** | 系统状态类 | -CPU:double  -RAM:double  -ROM:double | 同上 |

## 类图



# 接口设计

## 外部接口

本系统只针对“省人力资源数据采集系统”使用，不与其他系统产生交互，因此不存在与外部的接口。

## 内部接口

本系统主要分为四个层次DAO(数据持久层),Service(业务逻辑层),Action(控制层),表现层。内部接口是通过不同层之间的引用实现的，主要表现为业务逻辑层引用数据持久层以访问数据库；控制层引用业务逻辑层来完成主要业务逻辑操作，表现层引用控制层以实现控件等的事件处理。

# 出错处理设计

由于用户的不合法操作而出错时会提示错误信息，指引用户按一定的步骤进行合法操作。同时跳转到指定的错误处理页面或弹出出错信息对话框。

若服务器或网络发生错误，提示错误信息。

# 安全保密设计

只有拥有账号和密码的用户才可登陆系统进行相关操作。用户所有的操作都将被记录在日志中。

在系统中不同用户只能操作角色允许的功能，只能访问控制角色允许的数据，从而控制合法用户的操作权限，避免系统在权限控制方面的漏洞导致越权操作，产生安全问题。