|  |  |  |
| --- | --- | --- |
| **文件状态**  **[ ] 草稿**  **[ ] 讨论稿**  **[**√ **] 正式发布** | **文档类型** | **软件详细设计说明书** |
| **文件标识** | **XLM151008** |
| **版 本** | **3.0** |
| **作 者** | **山东省人力资源市场数据采集系统项目组** |
| **完成日期** | **2016.3.30** |

山东省人力资源市场数据采集系统

详细设计说明书

文档修订

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **更改人** | **描述（注明修改的条款或页）** |
| 1.0 | 2016.3.21 | 陈勇 | 详细设计说明书第一版 |
| 2.0 | 2016.3.27 | 陈勇 | 详细设计说明书第二版 |
| 3.0 | 2016.3.30 | 陈勇 | 详细设计说明书第三版 |
|  |  |  |  |
|  |  |  |  |

批准人签字

|  |  |  |
| --- | --- | --- |
| **职务** | **姓名** | **日期** |
| (项目经理、客户) |  |  |
| 项目经理 | 宦成颖 | 2016.3.21 |
| 项目经理 | 宦成颖 | 2016.3.27 |
| 项目经理 | 宦成颖 | 2016.3.30 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**目录**

[1 引言 5](#_Toc435891933)

[1.1 编写目的 5](#_Toc435891934)

[1.2 背景 5](#_Toc435891935)

[1.3 参考资料 6](#_Toc435891936)

[1.4 术语定义及说明 6](#_Toc435891937)

[2 设计概述 6](#_Toc435891938)

[2.1 需求概述 6](#_Toc435891939)

[2.1.1 业务需求 6](#_Toc435891940)

[2.1.2 输入 6](#_Toc435891941)

[2.1.3 输出 6](#_Toc435891942)

[2.1.4 主要功能 7](#_Toc435891943)

[2.1.5 性能 7](#_Toc435891944)

[2.2 运行环境概述 7](#_Toc435891945)

[2.3 条件与限制 9](#_Toc435891946)

[2.3.1 对用户手机的要求 9](#_Toc435891947)

[2.3.2 设计约束 9](#_Toc435891948)

[2.3.3 开发环境 9](#_Toc435891949)

[2.3.4 第三方软件 9](#_Toc435891950)

[2.3.5 网络设备 9](#_Toc435891951)

[2.3.6 技术限制 9](#_Toc435891952)

[2.3.7 进度限制 9](#_Toc435891953)

[3 总体方案确认 9](#_Toc435891954)

[3.1 系统详细界面划分 9](#_Toc435891955)

[4 系统详细设计 1](#_Toc435891956)0

[4.1 系统结构设计及子系统划分 10](#_Toc435891957)

[4.2 系统功能模块详细设计 11](#_Toc435891958)

[5 数据库系统设计](#_Toc435891995) 34

[5.1设计要求](#_Toc435891996) 34

[5.2 数据字典 3](#_Toc435891997)4

**1引言**

**1.1编写目的**

本详细设计说明书目的是对该项目进行详细设计，在概要设计的基础上进一步明确系统结构以及数据库系统设计，为进行后面的实现和测试作准备。本详细设计说明书的预期读者为本项目小组的成员以及对该系统感兴趣，在以后想对系统进行扩展和维护的人员。

**1.2背景**

项目名称：山东省人力资源市场数据采集系统

所属技术领域：移动互联网开发

承担部门：宦成颖的项目组

项目负责人：宦成颖

项目起止年限：2016年3月2日 至 2016年4月10日

系统主要目标：为了方便省内管理部门更好更快地掌握省内企业就业失业情况，及时地分析就业人数和失业人数改变的原因，以便做出合理的规划和整改措施，准确直观地做出图表图样，报给部委，作出决策。

**1.3参考资料**

需求规格说明书

概要设计说明书

**1.4术语定义及说明**

entity diagram:实体图

**2设计概述**

**2.1需求概述**

**2.1.1业务需求**

开发一个使用JSP和HTML语言，在各种主流浏览器下运行的一个网站，它的用户是企业和省级人员，他们具有一个上报企业数据、数据分享交流，一个审批、分析数据。

**2.1.2输入**

（1）如果调查期数据小于建档期数据，就业人数减少类型和就业人数减少主要原因及说明必填。

（2）就业人数减少类型包括：关闭破产、停业整顿、经济性裁员、业务转移、自然减员、正常解除或终止劳动合同、国际因素变化影响、自然灾害、重大事件影响、其他。

（3）就业人数减少原因包括：产业结构调整，重大技术改革，节能减排、淘汰落后产能，订单不足，原材料涨价，工资、社保等用工成本上升，自然减员，经营资金困难，税收政策变化（包括税负增加或出口退税减少等），季节性用工，其他，自行离职，工作调动、企业内部调剂，劳动关系转移、劳务派遣。

**2.1.3输出**

根据用户要求，选择企业数据，则输出企业数据表格。

**2.1.4主要功能**

（1）企业备案

查看所有已备案企业的详细信息，但不可以修改。显示所有已备案的企业，根据调查期和地区的查询条件进行检索，查看企业的详细信息，将当前的列表导出为EXCEL文件并保存在本地磁盘。

1. 企业报表管理

审核企业上报的数据并汇总上报到部级单位，查看企业上报的数据和报表。将企业上报的数据和报表退回修改，退回修改时可以添加备注，标识退回理由，将企业上报的数据和报表审核通过，将企业上报的数据上报到部级单位。

1. 对企业数据修改

可以对有明显错误的企业数据进行修改。修改后的数据另外存储并不修改报送的原始数据，对所选企业的数据进行修改，原始数据和修改日志保留。

1. 对企业数据删除

可以某一企业的数据进行删除，其对应企业的全部信息删除。

1. 对企业数据退回

将企业上报的数据进行退回，并提示对应企业。

1. 企业数据汇总

查看企业的汇总数据，根据不同的调查期显示出企业的汇总数据。

1. 企业数据导出

将某一企业导出为EXCEL文件并保存在本地磁盘。

1. 企业数据查询

对全省已创建账号进行条件查询，按照相应的查询条件进行查询，可以清除查询条件，查询条件有：单位名称、登陆账号、用户类型、所属地市、所属市县、所处区域、数据状态、单位性质、所属行业、起始日期、结束日期、统计月份、统计季度。

1. 数据多维分析

显示各市企业的数量和占比，全省各市企业的数量和占比以及相应的饼图，根据地区的查询条件进行检索。

1. 数据图表分析

1.对比分析

使用折线图、表格形式对比两个调查期的企业岗位变动情况，分析指标包括：企业总数、建档期总岗位数、调查期总岗位数、岗位变化总数、岗位减少总数、岗位变化数量占比。

2.趋势分析

使用折线图、表格形式展示多个连续调查期内的企业岗位变动情况，分析指标为：岗位变化数量占比。

1. 发布通知及浏览

发布、删除通知信息，列出当前用户发布的所有通知信息，可以新增、修改、删除通知，可供全部用户进行浏览。

1. 系统管理

1.上报时限

新增一个调查期或修改已有的调查期的时间。

2.用户管理

列出所有用户信息，新增一条用户信息，同时为该用户分配角色删除用户信息，如有该用户上报的数据，则不能删除，修改用户信息。

3.角色管理

定义新角色，分配功能权限，修改角色的功能权限，删除角色，如该角色已分配给用户，提示后删除角色和该角色与用户的关联关系，删除后需要为用户重新分配角色。

4.系统监控

查看当前系统工作情况。包括CPU、内存、硬盘等信息和应用系统的一些信息。

**2.1.5性能**

1. 对一次数据交换的响应时间（平均，最大）

平均1s, 最大2.5s

1. 数据交换的流量

个人50kb 左右，每秒最大10kb

1. 用户浏览容量

假设平均 1000， 最大 5000

1. 系统资源使用要求

内存控制在合理范围内

1. 兼容性

能兼容各大主流浏览器

**2.2运行环境概述**

1. 浏览器：IE、chrome、Firefox等主流浏览器
2. 操作系统：win系列、Linux、OSX
3. 数据库：Mysql
4. 硬件：手机或者电脑

**2.3条件与限制**

### 对用户手机、电脑的要求

现能用浏览器上网即可。

### 设计约束

1. 需要开启上网服务
2. 需要打开浏览器

### 开发环境

myeclipse ,myeclipse

### 第三方软件

无

### 网络设备

租用网络服务器

### 技术限制

模块间的衔接以及算法效率

### 进度限制

用户、开发、测试组间的跟进

**3总体方案确认**

**3.1系统详细界面划分**

**企业用户端**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **企业用户界面划分** | **覆盖范围** | **涉及的系统模块及方法** | **全局数据格式** | **系统性能要求** |
| 登录界面 | 登录功能 | 登录模块 | VARCHAR | Tomcat6.0及以上 |
| 企业信息修改 | 信息功能 | 信息模块 | VARCHAR | Tomcat6.0及以上 |
| 企业数据填报 | 信息功能 | 信息模块 | VARCHAR | Tomcat6.0及以上 |
| 企业数据查询 | 查询功能 | 查询模块 | VARCHAR | Tomcat6.0及以上 |

**市用户端**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **市用户界面划分** | **覆盖范围** | **涉及的系统模块及方法** | **全局数据格式** | **系统性能要求** |
| 审批上报 | 信息功能 | 信息模块 | VARCHAR | Tomcat6.0及以上 |
| 其余和省用户同 |  |  |  |  |

**省用户端**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **市用户界面划分** | **覆盖范围** | **涉及的系统模块及方法** | **全局数据格式** | **系统性能要求** |
| 企业备案 | 信息功能 | 信息模块 | VARCHAR | Tomcat6.0及以上 |
| 企业报表管理 | 信息功能 | 信息模块 | VARCHAR | Tomcat6.0及以上 |
| 对企业数据修改 | 信息功能 | 信息模块 | VARCHAR | Tomcat6.0及以上 |
| 对企业数据删除 | 信息功能 | 信息模块 | VARCHAR | Tomcat6.0及以上 |
| 对企业数据退回 | 信息功能 | 信息模块 | VARCHAR | Tomcat6.0及以上 |
| 企业数据汇总 | 信息功能 | 信息模块 | VARCHAR | Tomcat6.0及以上 |
| 企业数据导出 | 信息功能 | 信息模块 | VARCHAR | Tomcat6.0及以上 |
| 企业数据查询 | 查询功能 | 查询模块 | VARCHAR | Tomcat6.0及以上 |
| 数据多维分析 | 图表功能 | 图表模块 | VARCHAR | Tomcat6.0及以上 |
| 数据图表分析 | 图表功能 | 图表模块 | VARCHAR | Tomcat6.0及以上 |
| 发布信息及浏览 | 信息功能 | 信息模块 | VARCHAR | Tomcat6.0及以上 |
| 系统管理 | 系统功能 | 系统功能 | VARCHAR | Tomcat6.0及以上 |

**环境配置**

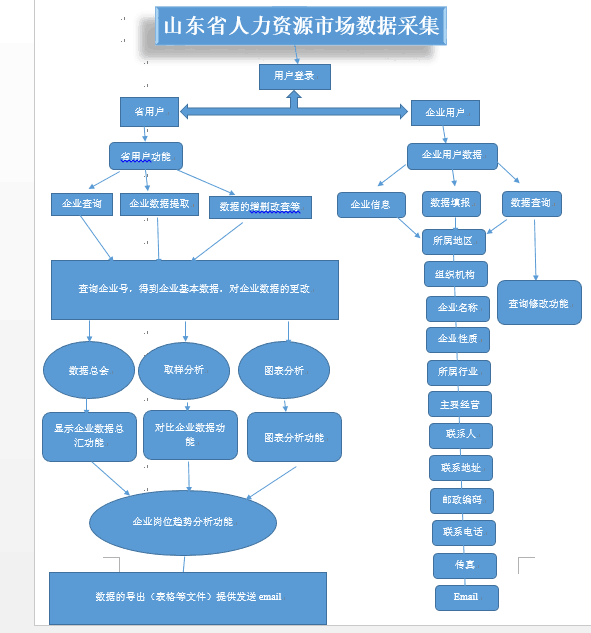
开发：Myeclipse

测试：Tomcat6.0及以上

运行：Tomcat6.0及以上

**4系统详细设计**

**4.1系统结构设计及子系统划分**



**4.2系统功能模块详细设计**

**4.2.1省用户**

## （1）数据退回

**接收接口：**

String table\_id = request.getParameter("table\_id");//table\_id企业tableID

**返回接口：**

HttpSession session = request.getSession();

session.setAttribute("flag\_ListProQueryEnt", "0");

response.sendRedirect("../province/List\_Manage.jsp");

**方法实现：**

**public** **boolean** datastatuschang(String table\_id, String n\_status) {

**try** {

*pstmt* = *conn*

.prepareStatement("update com\_data set status=? where table\_id=?");

*pstmt*.setString(1, n\_status);

*pstmt*.setString(2, table\_id);

*pstmt*.executeUpdate();

*pstmt* = *conn*

.prepareStatement("select com\_id from com\_data where table\_id=?");

*pstmt*.setString(1, table\_id);

ResultSet rs = *pstmt*.executeQuery();

rs.next();

String idString = rs.getString(1);

*pstmt* = *conn*

.prepareStatement("insert into news\_table(news\_head,news\_time,news\_content,news\_pub,news\_sub,news\_status) values('申请状态变更',now(),'您的申请状态已变更，请注意查看','0',?,'未查看')");

*pstmt*.setString(1, idString);

*pstmt*.executeUpdate();

**return** **true**;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **false**;

}

}

## （2）删除企业用户

**接收接口：**

String user\_id = request.getParameter("user\_id");

**返回接口：**

HttpSession session = request.getSession();

session.setAttribute("flagDelete", "1");

response.sendRedirect("../province/Sys\_usermanage.jsp");

**方法实现：**

**public** **void** deleteCityUser(String user\_id) {

**try** {

String senString = "delete from user\_table where user\_id = ?";

*pstmt* = *conn*.prepareStatement(senString);

*pstmt*.setString(1, user\_id);

*pstmt*.execute();

} **catch** (Exception e) {

// **TODO**: handle exception

}

}

## （3）查询企业报表

**接收接口：**

String table\_id = request.getParameter("table\_id");//table\_id企业tableID

**返回接口：**

HttpSession session = request.getSession();

session.setAttribute("oneQueryInfoList", enterpriseInfoList);

response.sendRedirect("../province/List\_ManageById.jsp");

**方法实现：**

## （4）数据退回

**接收接口：**

String table\_id = request.getParameter("table\_id");//table\_id企业tableID

**返回接口：**

HttpSession session = request.getSession();

session.setAttribute("flag\_ListProQueryEnt", "0");

response.sendRedirect("../province/List\_Manage.jsp");

**方法实现：**

**public** listManageTable queryOneListManage(String table\_id) {

**try** {

String sens = "select a.\*,b.\*,c.\* from com\_data a,time\_table b,com\_info c where a.table\_id = ? and a.time\_id = b.time\_id and a.com\_id = c.com\_id";

*pstmt* = *conn*.prepareStatement(sens);

*pstmt*.setString(1, table\_id);

ResultSet rs = *pstmt*.executeQuery();

listManageTable ent = **new** listManageTable();

**while** (rs.next()) {

ent.setCom\_id(rs.getString(1));

ent.setCom\_name(rs.getString(20));

ent.setCom\_area(rs.getString(19));

ent.setPeople\_ago(rs.getInt(3));

ent.setPeople\_now(rs.getInt(4));

ent.setOther\_reason(rs.getString(5));

ent.setType(rs.getString(7));

ent.setReason\_1(rs.getString(8));

ent.setExplain\_1(rs.getString(11));

ent.setReason\_2(rs.getString(9));

ent.setExplain\_2(rs.getString(12));

ent.setReason\_3(rs.getString(10));

ent.setExplain\_3(rs.getString(13));

}

**return** ent;

} **catch** (SQLException e) {

e.printStackTrace();

**return** **null**;

}

}

## （5）按地区查询企业

**接收接口：**

String com\_area = request.getParameter("area");// 地区

**返回接口：**

HttpSession session = request.getSession();

session.setAttribute("enterpriseInfoList", enterpriseInfoList);

session.setAttribute("flag\_ProQueryEnt", "1");

response.sendRedirect("../province/Enterprise\_Record.jsp");

**方法实现：**

**public** List selectEnterpriseDate(String place)// 根据地点查询企业信息

{

List list\_ent = **new** ArrayList();

**try** {

*pstmt* = *conn*

.prepareStatement("select com\_name,com\_area,com\_id from com\_info where com\_area=?");

*pstmt*.setString(1, place);

ResultSet rs = *pstmt*.executeQuery();

**while** (rs.next()) {

EnterpriseInfoTable enterpriseInfoTable = **new** EnterpriseInfoTable();

enterpriseInfoTable.setCom\_name(rs.getString(1));

enterpriseInfoTable.setCom\_area(rs.getString(2));

enterpriseInfoTable.setCom\_id(rs.getString(3));

list\_ent.add(enterpriseInfoTable);

}

**return** list\_ent;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **null**;

}

}

## （6）按照id查询企业信息

**接收接口：**

String Com\_id = request.getParameter("Com\_id");

**返回接口：**

EnterpriseInfoTable entReturn = **new** serviceOfPrince2().queryEnterpriseRecordById(Com\_id);

HttpSession session = request.getSession();

session.setAttribute("entReturnProQuery", entReturn);

response.sendRedirect("../province/Enterprise\_Record\_ById.jsp");

**方法实现：**

**public** EnterpriseInfoTable queryEnterpriseRecordById(String Com\_id) { // 查看具体企业详情

**try** {

*pstmt* = *conn*

.prepareStatement("select \* from com\_info where com\_id=?");

*pstmt*.setString(1, Com\_id);

// System.out.println("333222111");

ResultSet rs = *pstmt*.executeQuery();

EnterpriseInfoTable ent = **new** EnterpriseInfoTable();

**while** (rs.next()) {

ent.setCom\_id(rs.getString(1));

ent.setCom\_area(rs.getString(2));

ent.setCom\_name(rs.getString(3));

ent.setCom\_property(rs.getString(4));

ent.setCom\_industry(rs.getString(5));

ent.setCom\_business(rs.getString(6));

ent.setCom\_people(rs.getString(7));

ent.setCom\_address(rs.getString(8));

ent.setCom\_postalcode(rs.getString(9));

ent.setCom\_fax(rs.getString(10));

ent.setCom\_tel(rs.getString(11));

ent.setCom\_email(rs.getString(12));

}

**return** ent;

} **catch** (SQLException e) {

e.printStackTrace();

**return** **null**;

}

}

## （7）查询企业报表

**接收接口：**

String com\_id = request.getParameter("com\_id");// 企业ID

String com\_name = request.getParameter("com\_name");// 企业名称

String com\_year = request.getParameter("com\_year");// 年

String com\_month = request.getParameter("com\_month");// 月

**返回接口：**

HttpSession session = request.getSession();

session.setAttribute("listManageInfoList", listManageInfoList);

session.setAttribute("flag\_ListProQueryEnt", "1");

response.sendRedirect("../province/List\_Manage.jsp");

**方法实现：**

**public** List queryForm(String com\_name, String com\_id, String time\_year,

String time\_month) { // 报表管理\_查询

List list\_ent = **new** ArrayList();

// System.out.println("%%%%" + com\_name + "####" + com\_id + "####"

// + time\_year + "####" + time\_month);

**try** {

String sens = "select a.com\_name,a.com\_area,b.\*,c.time\_year,c.time\_month from com\_info a,com\_data b,time\_table c where b.status = '已上报' and a.com\_id = b.com\_id and c.time\_year=? and c.time\_month=? and c.time\_id = b.time\_id";

**int** cc1 = 0, cc2 = 0;

**if** (com\_name != "") {

sens = sens + " and a.com\_name=?";

cc1 = 1;

}

**if** (com\_id != "") {

sens = sens + " and b.com\_id=?";

cc2 = 1;

}

*pstmt* = *conn*.prepareStatement(sens);

*pstmt*.setString(1, time\_year);

*pstmt*.setString(2, time\_month);

**if** (cc1 == 1) {

**if** (cc2 == 1) {

*pstmt*.setString(3, com\_name);

*pstmt*.setString(4, com\_id);

} **else** {

*pstmt*.setString(3, com\_name);

}

} **else** {

**if** (cc2 == 1) {

*pstmt*.setString(3, com\_id);

}

}

ResultSet rs = *pstmt*.executeQuery();

**while** (rs.next()) {

listManageTable ent = **new** listManageTable();

// System.out.println("@@@" + ent.getCom\_name() + "####"

// + ent.getCom\_id() + "####");

ent.setCom\_name(rs.getString(1));

ent.setCom\_id(rs.getString(3));

ent.setCom\_area(rs.getString(2));

ent.setTable\_id(rs.getInt(4));

ent.setTime\_year(rs.getString(17));

ent.setTime\_month(rs.getString(18));

list\_ent.add(ent);

}

**return** list\_ent;

} **catch** (SQLException e) {

e.printStackTrace();

**return** **null**;

}

}

1. **插入消息**

方法：boolean insertnews(news\_table news)

说明：插入消息

输入：news\_table型数据 包括news\_table javabean中的全部六个属性

**private** String news\_head;//消息标题

**private** String news\_time;//发布时间

**private** String news\_content;//消息内容

**private** String news\_pub;//发布人ID

**private** String news\_sub;//接收人ID

**private** String news\_status;//消息状态（已查看/未查看）

返回：boolean型 返回插入是否成功（true or false）

实现方法：

**public** **boolean** insertnews(news\_table news) {// 消息插入

**try** {

pstmt = conn

.prepareStatement("insert into news\_table(news\_head,news\_time,news\_content,news\_pub,news\_sub,news\_status) values(?,?,?,?,?,?)");

pstmt.setString(1, news.getNews\_head());

pstmt.setString(2, news.getNews\_time());

pstmt.setString(3, news.getNews\_content());

pstmt.setString(4, news.getNews\_pub());

pstmt.setString(5, news.getNews\_sub());

pstmt.setString(6, news.getNews\_status());

pstmt.executeUpdate();

**return** **true**;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **false**;

}

}

1. **更改信息**

方法：boolean updatenews(news\_table news, String head)

说明：更改消息

输入：（1）news\_table型数据（同上） 包括news\_table javabean中的全部六个属性。

（2）string型数据 被更改的消息的标题。

返回：boolean型 返回更改是否成功（true or false）

实现方法：

**public** **boolean** updatenews(news\_table news, String head) {// 修改news

// 输入被修改的标题和新的news

**try** {

pstmt = conn

.prepareStatement("update news\_table set news\_head=?,news\_time=?,news\_content=?,news\_pub=?,news\_sub=?,news\_status=? where news\_head=?");

pstmt.setString(1, news.getNews\_head());

pstmt.setString(2, news.getNews\_time());

pstmt.setString(3, news.getNews\_content());

pstmt.setString(4, news.getNews\_pub());

pstmt.setString(5, news.getNews\_sub());

pstmt.setString(6, news.getNews\_status());

pstmt.setString(7, head);

pstmt.executeUpdate();

**return** **true**;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **false**;

}

}

1. **删除消息**

方法：boolean deletnews(String head, String time)

说明：删除消息

输入：string型数据 包括要删除的消息的标题和时间（时间格式待定）

返回：boolean型 返回插入是否成功（true or false）

实现方法：

**public** **boolean** deletnews(String head, String time) {// 删除news 输入标题和时间

**try** {

pstmt = conn

.prepareStatement("delete from news\_table where news\_head=? and news\_time=?");

pstmt.setString(1, head);

pstmt.setString(2, time);

pstmt.executeUpdate();

**return** **true**;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **false**;

}

}

**（11）查询某人发布的所有信息**

方法：ResultSet selectnewspub(String ID)

说明：查询某人发布的所有消息

输入：string型数据 输入发布人的ID

返回：ResultSet型数据 返回查询结果，顺序为news\_head,news\_time,news\_content,news\_pub,news\_sub,news\_status

实现方法：

**public** ResultSet selectnewspub(String ID) {// 发布者查询自己发布的消息

**try** {

pstmt = conn

.prepareStatement("select news\_head,news\_time,news\_content,news\_pub,news\_sub,news\_status from news\_table where news\_pub=?");

pstmt.setString(1, ID);

ResultSet rs = pstmt.executeQuery();

**return** rs;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **null**;

}

}

**（12）多条件查询企业申报的数据**

方法：ResultSet selectentrprisedata(PovinceDateQuery pdq)

说明：多条件查询企业申报的数据

输入：PovinceDateQuery型数据 输入类PovinceDateQuery中的全部属性

**private** String com\_name;

**private** String ID;

**private** String user\_type;

**private** String area;

**private** String com\_property;

**private** String start\_m;

**private** String start\_y;

**private** String end\_m;

**private** String end\_y;

其中 后四个为必填（用户没填就给我一个默认值），前五个选填（用户没填，就给我空字符串：””）

返回：ResultSet型数据 返回查询结果，顺序为com\_name,people\_ago,people\_now,other\_reson,type

实现方法：

**public** ResultSet selectentrprisedata(PovinceDateQuery pdq) {

String sql = "select com\_name,people\_ago,people\_now,other\_reson,type from com\_data ,com\_info where com\_data.com\_id=com\_info.com\_id and time\_id in(select time\_id from time\_table where (time\_year>=? or (time\_year=? and time\_month>=?))and(time\_year<=? or (time\_year=? and time\_month<=?)))";

**int** yearint1, yearint2;

String yearString1, yearString2;

yearint1 = Integer.*valueOf*(pdq.getStart\_y());

yearint1 += 1;

yearString1 = String.*valueOf*(yearint1);

yearint2 = Integer.*valueOf*(pdq.getEnd\_y());

yearint2 -= 1;

yearString2 = String.*valueOf*(yearint2);

**if** (!pdq.getCom\_name().equals("")) {

sql += "and com\_name=" + pdq.getCom\_name();

}

**if** (!pdq.getCom\_property().equals("")) {

sql += "and com\_property like '%" + pdq.getCom\_property() + "%'";

}

**if** (!pdq.getCom\_industry().equals("")) {

sql += "and com\_industry like '%" + pdq.getCom\_industry() + "%'";

}

**if** (!pdq.getArea().equals("")) {

sql += "and com\_area=" + pdq.getArea();

}

**if** (!pdq.getID().equals("")) {

sql += "and com\_info.com\_id=" + pdq.getID();

}

**if** (!pdq.getUser\_type().equals("")) {

sql += "and com\_info.com\_id in(select user\_id from user\_table where user\_type="

+ pdq.getUser\_type() + ")";

}

**try** {

System.***out***.println(sql);

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, yearString1);

pstmt.setString(2, pdq.getStart\_y());

pstmt.setString(3, pdq.getStart\_m());

pstmt.setString(4, yearString2);

pstmt.setString(5, pdq.getEnd\_y());

pstmt.setString(6, pdq.getEnd\_m());

ResultSet rs = pstmt.executeQuery();

**return** rs;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **null**;

}

}

**（13）根据当前调查期查询（表格以及饼状图）**

方法：ResultSet tableviewenterprise(String year,String month)

说明：根据当前调查期查询（表格以及饼状图）

输入：string型数据 包括当前调查期的年和月

返回：返回：ResultSet型数据 返回查询结果，顺序为：com\_area（地区）,count(com\_info.com\_id)（该地区的企业数）,sum(people\_now-people\_ago)（人数改变的总和）

实现方法：

**public** ResultSet tableviewenterprise(String year, String month) {

**try** {

pstmt = conn

.prepareStatement("select com\_area,count(com\_info.com\_id),sum(people\_now-people\_ago) from com\_data,com\_info where com\_data.com\_id=com\_info.com\_id and time\_id in(select time\_id from time\_table where time\_year=? and time\_month=?)group by com\_area");

pstmt.setString(1, year);

pstmt.setString(2, month);

ResultSet rs = pstmt.executeQuery();

**return** rs;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **null**;

}

}

**（14）返回符合查询条件的所有企业在某一调查期的雇佣人数和**

方法：int allpeople\_nowselect(String year,String month,String property,String industry,String area)

说明： 返回符合查询条件的所有企业在某一调查期的雇佣人数和。

输入：5个string型，调查期的年、月，企业性质，行业，地区。

年月必填，后三个可以为空字符串:””

返回：int型 返回人数总和。

实现方法：

**public** **int** allpeople\_nowselect(String year, String month, String property,

String industry, String area) {

String sql = "select sum(com\_data.people\_now) from com\_data,com\_info where com\_data.time\_id in(select time\_id from time\_table where time\_year=? and time\_month=?) and com\_data.com\_id=com\_info.com\_id";

**if** (!property.equals("")) {

sql += "and com\_property=" + property;

}

**if** (!industry.equals("")) {

sql += "and com\_industry=" + industry;

}

**if** (!area.equals("")) {

sql += "and com\_area=" + area;

}

**int** number = 0;

**try** {

pstmt = conn.prepareStatement(sql);

pstmt.setString(1, year);

pstmt.setString(2, month);

ResultSet rs = pstmt.executeQuery();

rs.next();

number = Integer.*valueOf*(rs.getString(1));

**return** number;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** 0;

}

}

**（15）返回某一企业一段时间内的雇佣人数**

方法：ResultSet onepeople\_nowselect(String s\_y,String s\_m,String e\_y,String e\_m,String com\_name)

说明：返回某一企业一段时间内的雇佣人数

输入：五个string型 包括起始调查期的年、月，结束调查期的年、月，企业名称。

返回：ResultSet型 包含一列，表示调查期的雇佣人数。

实现方法：

**public** ResultSet onepeople\_nowselect(String s\_y, String s\_m, String e\_y,

String e\_m, String com\_name) {

**try** {

pstmt = conn

.prepareStatement("select sun(people\_now) from com\_data ,com\_info where com\_name=? and com\_data.com\_id=com\_info.com\_id and time\_id in(select time\_id from time\_table where (time\_year>=? or (time\_year=? and time\_month>=?))and(time\_year<=? or (time\_year=? and time\_month<=?)))");

String year = "";

**int** yearint = 0;

pstmt.setString(1, com\_name);

yearint = Integer.*valueOf*(s\_y);

yearint++;

year = String.*valueOf*(yearint);

pstmt.setString(2, year);

pstmt.setString(3, s\_y);

pstmt.setString(4, s\_m);

yearint = Integer.*valueOf*(e\_y);

yearint--;

year = String.*valueOf*(yearint);

pstmt.setString(5, year);

pstmt.setString(6, e\_y);

pstmt.setString(7, e\_m);

ResultSet rsResultSet = pstmt.executeQuery();

**return** rsResultSet;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **null**;

}

}

}

# 4.2.2市用户

## （1）数据退回

**接收接口：**

String table\_id = request.getParameter("table\_id");// 企业ID

**返回接口：**

HttpSession session = request.getSession();

session.setAttribute("flag\_ListProQueryEnt", "0");

response.sendRedirect("../city/City\_ListManage.jsp");

**方法实现：**

**public** **boolean** datastatuschangForCity(String table\_id, String n\_status) { // 退回

**try** {

*pstmt* = *conn*

.prepareStatement("update com\_data set status=? where table\_id=?");

*pstmt*.setString(1, n\_status);

*pstmt*.setString(2, table\_id);

*pstmt*.executeUpdate();

*pstmt* = *conn*

.prepareStatement("select com\_id from com\_data where table\_id=?");

*pstmt*.setString(1, table\_id);

ResultSet rs = *pstmt*.executeQuery();

rs.next();

String idString = rs.getString(1);

*pstmt* = *conn*

.prepareStatement("insert into news\_table(news\_head,news\_time,news\_content,news\_pub,news\_sub,news\_status) values('申请状态变更',now(),'您的申请状态已变更，请注意查看','0',?,'未查看')");

*pstmt*.setString(1, idString);

*pstmt*.executeUpdate();

**return** **true**;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **false**;

}

}

## （2）数据上报

**接收接口：**

String table\_id = request.getParameter("table\_id");// 企业ID

**返回接口：**

HttpSession session = request.getSession();

session.setAttribute("flag\_ListProQueryEnt", "0");

response.sendRedirect("../city/City\_ListManage.jsp");

**方法实现：**

**public** **boolean** dataStatusGoonForCity(String table\_id, String n\_status) { // 上报

**try** {

*pstmt* = *conn*

.prepareStatement("update com\_data set status=? where table\_id=?");

*pstmt*.setString(1, n\_status);

*pstmt*.setString(2, table\_id);

*pstmt*.executeUpdate();

*pstmt* = *conn*

.prepareStatement("select com\_id from com\_data where table\_id=?");

*pstmt*.setString(1, table\_id);

ResultSet rs = *pstmt*.executeQuery();

rs.next();

String idString = rs.getString(1);

*pstmt* = *conn*

.prepareStatement("insert into news\_table(news\_head,news\_time,news\_content,news\_pub,news\_sub,news\_status) values('申请状态变更',now(),'您的申请状态已变更，请注意查看','0',?,'未查看')");

*pstmt*.setString(1, idString);

*pstmt*.executeUpdate();

**return** **true**;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **false**;

}

}

## （3）查询企业报表

**接收接口：**

HttpSession session0 = request.getSession();

String user\_id = (String) session0.getAttribute("user");

String table\_id = request.getParameter("table\_id");// 企业table\_ID

**返回接口：**

HttpSession session = request.getSession();

session.setAttribute("cityInfoList", list);

session.setAttribute("flag\_ProQueryEnt", "1");

response.sendRedirect("../city/City\_ListManageById.jsp");

**方法实现：**

**public** listManageTable queryOneListManageForCity(String table\_id,

String user\_id) {

**try** {

*pstmt* = *conn*

.prepareStatement("select com\_area from com\_info where com\_id = ?");

*pstmt*.setString(1, user\_id);

ResultSet rs0 = *pstmt*.executeQuery();

String cityString = "";

**if** (rs0.first()) {

cityString = rs0.getString(1);

}

String sens = "select a.\*,b.\*,c.\* from com\_data a,time\_table b,com\_info c where a.table\_id = ? and a.time\_id = b.time\_id and a.com\_id = c.com\_id and c.com\_area = '"

+ cityString + "'";

*pstmt* = *conn*.prepareStatement(sens);

*pstmt*.setString(1, table\_id);

ResultSet rs = *pstmt*.executeQuery();

listManageTable ent = **new** listManageTable();

**while** (rs.next()) {

ent.setCom\_id(rs.getString(1));

ent.setCom\_name(rs.getString(20));

ent.setCom\_area(rs.getString(19));

ent.setPeople\_ago(rs.getInt(3));

ent.setPeople\_now(rs.getInt(4));

ent.setOther\_reason(rs.getString(5));

ent.setType(rs.getString(7));

ent.setReason\_1(rs.getString(8));

ent.setExplain\_1(rs.getString(11));

ent.setReason\_2(rs.getString(9));

ent.setExplain\_2(rs.getString(12));

ent.setReason\_3(rs.getString(10));

ent.setExplain\_3(rs.getString(13));

}

**return** ent;

} **catch** (SQLException e) {

e.printStackTrace();

**return** **null**;

}

}

## （4）查询企业信息

**接收接口：**

String Com\_id = request.getParameter("Com\_id");

**返回接口：**

HttpSession session = request.getSession();

session.setAttribute("entReturnProQuery", entReturn);

response.sendRedirect("../city/City\_Record\_ById.jsp");

**方法实现：**

**public** EnterpriseInfoTable queryEnterpriseRecordByIdForEnterpriseInfoTableCity(

String Com\_id) { // 查看具体企业详情

**try** {

*pstmt* = *conn*

.prepareStatement("select \* from com\_info where com\_id=?");

*pstmt*.setString(1, Com\_id);

// System.out.println("333222111");

ResultSet rs = *pstmt*.executeQuery();

EnterpriseInfoTable ent = **new** EnterpriseInfoTable();

**while** (rs.next()) {

ent.setCom\_id(rs.getString(1));

ent.setCom\_area(rs.getString(2));

ent.setCom\_name(rs.getString(3));

ent.setCom\_property(rs.getString(4));

ent.setCom\_industry(rs.getString(5));

ent.setCom\_business(rs.getString(6));

ent.setCom\_people(rs.getString(7));

ent.setCom\_address(rs.getString(8));

ent.setCom\_postalcode(rs.getString(9));

ent.setCom\_fax(rs.getString(10));

ent.setCom\_tel(rs.getString(11));

ent.setCom\_email(rs.getString(12));

}

**return** ent;

} **catch** (SQLException e) {

e.printStackTrace();

**return** **null**;

}

}

## （5）检索出某调查期内所有上报的企业

**接收接口：**

HttpSession session0 = request.getSession();

String user\_id = (String) session0.getAttribute("user");

String com\_id = request.getParameter("com\_id");// 企业ID

String com\_name = request.getParameter("com\_name");// 企业名称

String com\_year = request.getParameter("com\_year");// 年

String com\_month = request.getParameter("com\_month");// 月

**返回接口：**

HttpSession session = request.getSession();

session.setAttribute("listManageInfoList", listManageInfoList);

session.setAttribute("flag\_ListProQueryEnt", "1");

response.sendRedirect("../city/City\_ListManage.jsp");

**方法实现：**

HttpSession session = request.getSession();

session.setAttribute("listManageInfoList", listManageInfoList);

session.setAttribute("flag\_ListProQueryEnt", "1");

response.sendRedirect("../city/City\_ListManage.jsp");

# 4.2.3企业用户

## （1）企业数据查询

**接收接口：**

String com\_id = ((userInfoTable) request.getAttribute("user")).getUser\_id();//获取企业id,作为组织机构的com\_id

**返回接口：**

EnterpriseDataTable enterpriseInfoTable = **new** serviceOfEnterprise().queryEnterpriseData(com\_id, time\_year, time\_month);

HttpSession session = request.getSession();

session.setAttribute("enterpriseInfoTable", enterpriseInfoTable);

**方法实现：**

**public** EnterpriseDataTable queryEnterpriseData(String com\_id,String time\_year,String time\_month) { //查询企业数据填报

EnterpriseDataTable ent = **new** EnterpriseDataTable();

**try** {

pstmt = conn

.prepareStatement("select \* from com\_data a,time\_table b where a.com\_id=？ and a.time\_id=b.time\_id and b.time\_year=？ and b.time\_month=？");

pstmt.setString(1, com\_id);

pstmt.setString(2, time\_year);

pstmt.setString(3, time\_month);

ResultSet rs = pstmt.executeQuery();

**while** (rs.next()) {

ent.setPeople\_ago(Integer.*valueOf*(rs.getString(3)));

ent.setPeople\_now(Integer.*valueOf*(rs.getString(4)));

ent.setOther\_reason(rs.getString(5));

ent.setType(rs.getString(7));

ent.setReason\_1(rs.getString(8));

ent.setReason\_2(rs.getString(9));

ent.setReason\_3(rs.getString(10));

ent.setExplain\_1(rs.getString(11));

ent.setExplain\_2(rs.getString(12));

ent.setExplain\_3(rs.getString(13));

}

**return** ent;

} **catch** (SQLException e) {

e.printStackTrace();

**return** **null**;

}

}

## （2）修改企业信息

**接收接口：**

String com\_id = ((userInfoTable) request.getAttribute("user")).getUser\_id();//获取企业id,作为组织机构的com\_id

**返回接口：**

HttpSession session = request.getSession();

**if**(**new** serviceOfEnterprise().modifyEnterpriseInfo(enterpriseInfoTable)){

session.setAttribute("message", "success");//如果新增数据成功，则封装一个成功的Session信号

}

**else**{

session.setAttribute("message", "failed");//如果新增数据失败，则封装一个失败的Session信号

}

response.sendRedirect("../EnterpriseInfo.jsp");//跳转回原界面

**方法实现：**

**public** **boolean** modifyEnterpriseInfo(EnterpriseInfoTable ent) { //修改企业信息

**try** {

pstmt = conn

.prepareStatement("update com\_info set com\_area=? , com\_name=? ,com\_ property=? ,"

+ " com\_ industry=? ,com\_ business=? , com\_people=?,com\_address=? , com\_postalcode=? ,"

+ "com\_fax=? ,com\_tel=? ,com\_email=? where com\_id=?");

pstmt.setString(1, ent.getCom\_area());

pstmt.setString(2, ent.getCom\_name());

pstmt.setString(3, ent.getCom\_property());

pstmt.setString(4, ent.getCom\_industry());

pstmt.setString(5, ent.getCom\_business());

pstmt.setString(6, ent.getCom\_people());

pstmt.setString(7, ent.getCom\_address());

pstmt.setString(8, ent.getCom\_postalcode());

pstmt.setString(9, ent.getCom\_fax());

pstmt.setString(10, ent.getCom\_tel());

pstmt.setString(11, ent.getCom\_email());

pstmt.setString(12, ent.getCom\_id());

pstmt.executeUpdate();

**return** **true**;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **false**;

}

}

## （3）数据上报

**接收接口：**

Integer people\_ago = Integer.*valueOf*(request.getParameter("people\_ago"));

Integer people\_now = Integer.*valueOf*(request.getParameter("people\_now"));

String other\_reason = request.getParameter("other\_reason");

String type = request.getParameter("type");

String reason\_1 = request.getParameter("reason\_1");

String reason\_2 = request.getParameter("reason\_2");

String reason\_3 = request.getParameter("reason\_3");

String explain\_1 = request.getParameter("explain\_1");

String explain\_2 = request.getParameter("explain\_2");

String explain\_3 = request.getParameter("explain\_3");

String status = request.getParameter("status");

Integer time\_id = Integer.*valueOf*(request.getParameter("time\_id"));

EnterpriseDataTable enterpriseDataTable = **new** EnterpriseDataTable();

**返回接口：**

HttpSession session = request.getSession();

**if**(**new** serviceOfEnterprise().addEnterpriseData(enterpriseDataTable, ((userInfoTable) request.getAttribute("user")).getUser\_id())){

session.setAttribute("message", "success");//如果新增数据成功，则封装一个成功的Session信号

}

**else**{

session.setAttribute("message", "failed");//如果新增数据失败，则封装一个失败的Session信号

}

**方法实现：**

**public** **boolean** addEnterpriseData(EnterpriseDataTable ent,String com\_id) { //填报企业数据

**try** {

pstmt = conn.prepareStatement("insert into com\_data"

+ "(com\_id,people\_ago,people\_now,other\_reson,time\_id,type,reason\_1,reason\_2,reason\_3,explain\_1,explain\_2,explain\_3,status) "

+ "values(?,?,?,?,?,?,?,?,?,?,?,?,?)");

pstmt.setString(1, com\_id);

pstmt.setString(2, ent.getPeople\_ago().toString());//后面改的-yu

pstmt.setString(3, ent.getPeople\_now().toString());

pstmt.setString(4, ent.getOther\_reason());

pstmt.setString(5, ent.getTime\_id().toString());

pstmt.setString(6, ent.getType());

pstmt.setString(7, ent.getReason\_1());

pstmt.setString(8, ent.getReason\_2());

pstmt.setString(9, ent.getReason\_3());

pstmt.setString(10, ent.getExplain\_1());

pstmt.setString(11, ent.getReason\_2());

pstmt.setString(12, ent.getReason\_3());

pstmt.setString(13, ent.getStatus());

pstmt.executeUpdate();

**return** **true**;

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

**return** **false**;

}

}

**5数据库系统设计**

**5.1设计要求**

（1）满足用户要求：既能合理地组织用户需要的所有数据，又能支持用户对数据的所有处理功能。

（2）满足某个数据库管理系统的要求：能够在数据库管理系统中实现。

（3）具有较高的范式：数据完整性好、效益高，便于理解和维护，没有数据冲突。

**5.2 数据字典**

1. 企业用户信息表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 长度 | 键 | null | 示例 |
| com\_id | 组织机构代码 | varchar | 20 | PK |  | 123 |
| com\_area | 所属地区 | varchar | 30 |  |  | 济南 |
| com\_name | 企业名称 | varchar | 30 |  |  | 能源科技 |
| com\_property | 企业性质 | varchar | 20 |  |  | 见下文说明 |
| com\_industry | 所属行业 | varchar | 20 |  |  | 科技 |
| com\_business | 主要经营业务 | varchar | 100 |  |  | 扫描仪 |
| com\_people | 联系人 | varchar | 30 |  |  | 马云 |
| com\_address | 地址 | varchar | 100 |  |  | 中关村南大街5号 |
| com\_postalcode | 邮政编码 | varchar | 10 |  |  | 100000 |
| com\_tel | 电话 | varchar | 20 |  |  | 略 |
| com\_fax | 传真 | varchar | 20 |  |  | 略 |
| com\_email | 邮箱 | varchar | 50 |  | yes | 略 |

1. 企业数据填报表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 长度 | 键 | null | 示例 |
| com\_id | 组织机构代码 | varchar | 20 | FK |  | 123 |
| table\_id | 数据表id | int |  | PK |  | 自增int，插入时自动生成 |
| people\_ago | 建档期就业人数 | int |  |  |  | 100 |
| people\_now | 调查期就业人数 | int |  |  |  | 200 |
| other\_reason | 其他原因 | varchar | 100 |  |  | 略 |
| type | 就业人数减少类型 | varchar | 20 |  | yes | 略 |
| reason\_1 | 主要原因 | varchar | 100 |  | yes | 关闭破产 |
| explain\_1 | 主要原因说明 | varchar | 100 |  | yes | 企业倒闭...... |
| reason\_2 | 次要原因 | varchar | 100 |  | yes |  |
| explain\_2 | 次要原因说明 | varchar | 100 |  | yes |  |
| reason\_3 | 第三原因 | varchar | 100 |  | yes |  |
| explain\_3 | 第三原因说明 | varchar | 100 |  | yes |  |
| status | 报表状态 | varchar | 50 |  |  |  |
| time\_id | 调查期id | int |  | FK |  | 1（外键，对应调查其编号） |

1. 调查期表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 长度 | 键 | null | 示例 |
| time\_id | 调查期id | int |  | PK |  | 自增整形，插入时自动生成 |
| time\_year | 调查期年份 | varchar | 20 |  |  | 2016 |
| time\_month | 调查期月份 | varchar | 20 |  |  | 1 |

1. 用户表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 长度 | 键 | null | 示例 |
| user\_id | 用户帐号 | varchar | 20 | PK |  | 231 |
| password | 用户密码 | varchar | 20 |  |  | 321 |
| user\_type | 用户类型 | varchar | 10 |  |  | enterprise/province（二选一） |
| user\_role | 用户角色 | varchar | 10 | FK |  | administrator |

1. 角色表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段 | 说明 | 类型 | 长度 | 键 | null | 示例 |
| user\_role | 用户角色 | varchar | 10 | PK |  | administrator |
| news\_pub | 消息发布 | varchar | 10 |  |  | 0或1（0代表无权限，1代表有权限） |
| news\_delete | 消息删除 | varchar | 10 |  |  | 0或1 |