NTT DATA

软件项目工程师奖金管理系统

(程序设计报告)

1.项目总则细则设定

主要对应以系统管理员登录权限的功能，包括项目总则设定，奖金标准设定，正式社员细则标准，连携部分细则设定，bp 部分细则设定，申请时已退职社员细则设定。

该模块总则设定业务层代码如下：

```
package com.ntt.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import com.ntt.model.PageBean;

import com.ntt.model.ZongZe;

import com.ntt.util.StringUtil;

public class ZongZeDao {

    /**

     * zongZe List method

     *

     * @param con

     * @param pageBean

     * @param s_zongZe

     * @return

     * @throws Exception

     */

    public List<ZongZe> zongZeList(Connection con, PageBean pageBean, ZongZe
s_zongZe) throws Exception {

        List<ZongZe> zongZesList = new ArrayList<ZongZe>();

        StringBuffer sb = new StringBuffer("SELECT * FROM t_zongze t1");

        if (StringUtil.isNotEmpty(s_zongZe.getCenterName())) {

            sb.append(" where t1.centerName like '%" + s_zongZe.getCenterName()
+ "%'");
```

```
        }
        if (pageBean != null) {
            sb.append("  limit  "  +  pageBean.getStart()  +  ","  +
pageBean.getPageSize());
        }
        PreparedStatement pstmt = con.prepareStatement(sb.toString());
        ResultSet rs = pstmt.executeQuery();
        while (rs.next()) {
            ZongZe zongZe = new ZongZe();
            zongZe.setCenterId(rs.getInt("centerId"));
            zongZe.setCenterName(rs.getString("centerName"));
            zongZe.setBpRate(rs.getString("bpRate"));
            zongZe.setProjectRateUp(rs.getString("projectRateUp"));
            zongZe.setProjectRateDown(rs.getString("projectRateDown"));
            zongZe.setShareBonusRateUp(rs.getString("shareBonusRateUp"));
            zongZe.setShareBonusRateDown(rs.getString("shareBonusRateDown"));
            zongZe.setTuijianRate(rs.getString("tuijianRate"));
            zongZesList.add(zongZe);
        }
        return zongZesList;
    }

    /**
     * zongze count function
     *
     * @param con
     * @param s_ZongZe
     * @return
     * @throws Exception
     */
    public int zongZeCount(Connection con, ZongZe s_ZongZe) throws Exception {
```

```
        StringBuffer sb = new StringBuffer("select count(*) as total from
t_zongze t1");
        if (StringUtil.isNotEmpty(s_ZongZe.getCenterName())) {
            sb.append(" where t1.centerName like '%" + s_ZongZe.getCenterName()
+ "%'");
        }
        PreparedStatement pstmt = con.prepareStatement(sb.toString());
        ResultSet rs = pstmt.executeQuery();
        if (rs.next()) {
            return rs.getInt("total");
        } else {
            return 0;
        }
    }

    /**
     * zongze show function
     *
     * @param con
     * @param centerId
     * @return
     * @throws Exception
     */
    public ZongZe zongZeShow(Connection con, String centerId) throws Exception
{
        String sql = "select * from t_zongze t1 where t1.centerId=?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, centerId);
        ResultSet rs = pstmt.executeQuery();
        ZongZe zongZe = new ZongZe();
        if (rs.next()) {
            zongZe.setCenterId(rs.getInt("centerId"));
```

```
            zongZe.setCenterName(rs.getString("centerName"));
            zongZe.setBpRate(rs.getString("bpRate"));
            zongZe.setProjectRateUp(rs.getString("projectRateUp"));
            zongZe.setProjectRateDown(rs.getString("projectRateDown"));
            zongZe.setShareBonusRateUp(rs.getString("shareBonusRateUp"));
            zongZe.setShareBonusRateDown(rs.getString("shareBonusRateDown"));
            zongZe.setTuijianRate(rs.getString("tuijianRate"));
        }
        return zongZe;
    }

    /**
     * ZONGZE ADD fUNCTION
     *
     * @param con
     * @param zongZe
     * @return
     * @throws Exception
     */
    public int zongZeAdd(Connection con, ZongZe zongZe) throws Exception {
        String sql = "insert into t_zongze values(null,?,?,?,?,?,?,?)";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, zongZe.getCenterName());
        pstmt.setString(2, zongZe.getBpRate());
        pstmt.setString(3, zongZe.getProjectRateUp());
        pstmt.setString(4, zongZe.getProjectRateDown());
        pstmt.setString(5, zongZe.getShareBonusRateUp());
        pstmt.setString(6, zongZe.getShareBonusRateDown());
        pstmt.setString(7, zongZe.getTuijianRate());
        return pstmt.executeUpdate();
    }
```

```
    /**
     * Delete Function
     *
     * @param con
     * @param centerId
     * @return
     * @throws Exception
     */
    public int zongZeDelete(Connection con, String centerId) throws Exception
{
        String sql = "delete from t_zongze where centerId=?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, centerId);
        return pstmt.executeUpdate();
    }


    /**
     * ZONGZE UPDATE FUNCTION
     *
     * @param con
     * @param zongze
     * @return
     * @throws Exception
     */
    public int zongZeUpdate(Connection con, ZongZe zongze) throws Exception {
        String sql = "update t_zongze set
centerName=?,bpRate=?,projectRateUp=?,projectRateDown=?,shareBonusRateUp=?,
shareBonusRateDown=?,tuijianRate=? where centerId=?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, zongze.getCenterName());
        pstmt.setString(2, zongze.getBpRate());
```

```
        pstmt.setString(3, zongze.getProjectRateUp());
        pstmt.setString(4, zongze.getProjectRateDown());
        pstmt.setString(5, zongze.getShareBonusRateUp());
        pstmt.setString(6, zongze.getShareBonusRateDown());
        pstmt.setString(7, zongze.getTuijianRate());
        pstmt.setInt(8, zongze.getCenterId());
        return pstmt.executeUpdate();
    }

    /**
     * serach data by centername
     *
     * @param con
     * @param name
     * @return
     * @throws Exception
     */
    public boolean haveCenterByCenterName(Connection con, String name) throws
Exception {
        String sql = "select * from t_zongze t1 where t1.centerName=?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, name);
        ResultSet rs = pstmt.executeQuery();
        if (rs.next()) {
            return true;
        }
        return false;
    }
}
```

2. 项目奖金总额申请

此模块的功能可以导入项目基本信息，并对应该项目申请奖金，其主要字段、类型等见下表 1-1。

	字段内容	类型	说明
1	申请 ID	字符型，10 位英字	Key：0000102779
2	财务 ID	字符型	
3	项目名称	字符型	项目名称信息可以从其他数据导入作为 Master 数据
4	ＰＭ员工 ID	字符型，6 位英字	注意姓名的匹配
5	部门 ＩＤ	字符型，10 位英字	注意部门名称的匹配
6	部门名称	字符型，50 位英字	
7	正式社员人月数	数值型，2 位小数	最大值 10000 即可
8	ＢＰ人月数	数值型，2 位小数	最大值 10000 即可
9	连携人月数	数值型，2 位小数	最大值 10000 即可
10	离职人月数	数值型，2 位小数	最大值 10000 即可
11	申请对象期间	字符型，150 字	
12	申请者 ＩＤ	字符型，6 位英字	
13	备注	字符型，150 字	
14	申请日期	日期型	

表 1-1 项目奖金总额申请字段表

实际操作页面可见图 1-1 及图 1-2

财务ID	项目名称	受注实绩	请求实绩	连携实绩	项目开始日	项目结束日	操作	奖金申请
5W211011605	项目1	50.0	50.0	40.0	2016-08-01	2016-08-02	修改更新 删除	奖金申请 查询
5W211011505	项目2	40.0	40.0	10.0	2016-07-01	2016-07-31	修改更新 删除	奖金申请 查询
3	3	3.0	3.0	3.0	2017-08-15	2017-08-15	修改更新 删除	奖金申请 查询

图 1-1 页面一

财务ID	项目名称	部门名称	申请人姓名	正式员工人月数	BP人月数	连携人月数	离职人月数	申请日期	奖金总额
5W211011605	项目1	2H1B	张三	40.0	10.0	40.0	0.0	1-3月	75000

图 1-2 导入信息页面

该模块的关键代码如下：

```
public class AppliBonusDao {


//列出所有项目

public List<AppliBonus> appliBonusList(Connection con, PageBean pageBean, AppliBonus s_AppliBonus) throws Exception {
        List<AppliBonus> appliBonusList = new ArrayList<AppliBonus>();
        StringBuffer sb = new StringBuffer("SELECT * FROM t_project t1");
        if (StringUtil.isNotEmpty(s_AppliBonus.getProjectName())) {
            sb.append("      where      t1.projectName      like      '%"      +
s_AppliBonus.getProjectName() + "%'");
        }
        if (pageBean != null) {
            sb.append("    limit    "    +    pageBean.getStart()    +    ","    +
pageBean.getPageSize());
        }
        PreparedStatement pstmt = con.prepareStatement(sb.toString());
        ResultSet rs = pstmt.executeQuery();
        while (rs.next()) {
            AppliBonus appliBonus = new AppliBonus();
            appliBonus.setProjectId(rs.getInt("projectId"));
            appliBonus.setFinancialId(rs.getString("financialId"));
            appliBonus.setProjectName(rs.getString("projectName"));
            appliBonus.setOrderPerformance(rs.getDouble("orderPerformance"));

    appliBonus.setRequestPerformance(rs.getDouble("requestPerformance"));
```

```
    appliBonus.setSupportPerformance(rs.getDouble("supportPerformance"));
            appliBonus.setProjectStartDate(rs.getDate("projectStartDate"));
            appliBonus.setProjectEndDate(rs.getDate("projectEndDate"));
            appliBonus.setCenterId(rs.getInt("centerId"));
            appliBonusList.add(appliBonus);
        }
        return appliBonusList;
    }


    public int appliBonusCount(Connection con, AppliBonus s_AppliBonus) throws
Exception {
        StringBuffer sb = new StringBuffer("select count(*) as total from
t_project t1");
        if (StringUtil.isNotEmpty(s_AppliBonus.getProjectName())) {
            sb.append("      where      t1.projectName      like      '%"      +
s_AppliBonus.getProjectName() + "%'");
        }
        PreparedStatement pstmt = con.prepareStatement(sb.toString());
        ResultSet rs = pstmt.executeQuery();
        if (rs.next()) {
            return rs.getInt("total");
        } else {
            return 0;
        }
    }
```

```
    public AppliBonus appliBonusShow(Connection con, String projectId) throws
Exception {
        String sql = "select * from t_project t1 where t1.projectId=?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, projectId);
        ResultSet rs = pstmt.executeQuery();
        AppliBonus appliBonus = new AppliBonus();
        if (rs.next()) {
            appliBonus.setProjectId(rs.getInt("projectId"));
            appliBonus.setFinancialId(rs.getString("financialId"));
            appliBonus.setProjectName(rs.getString("projectName"));
            appliBonus.setOrderPerformance(rs.getDouble("orderPerformance"));

    appliBonus.setRequestPerformance(rs.getDouble("requestPerformance"));

    appliBonus.setSupportPerformance(rs.getDouble("supportPerformance"));
            appliBonus.setProjectStartDate(rs.getDate("projectStartDate"));
            appliBonus.setProjectEndDate(rs.getDate("projectEndDate"));
        }
        return appliBonus;
    }

//添加新项目
    public int appliBonusAdd(Connection con, AppliBonus appliBonus) throws
Exception {
        String sql = "insert into t_project values(null,?,?,?,?,?,?,?,null)";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, appliBonus.getFinancialId());
        pstmt.setString(2, appliBonus.getProjectName());
```

```
        pstmt.setDouble(3, appliBonus.getOrderPerformance());
        pstmt.setDouble(4, appliBonus.getRequestPerformance());
        pstmt.setDouble(5, appliBonus.getSupportPerformance());
        pstmt.setDate(6,new
java.sql.Date(appliBonus.getProjectStartDate().getTime()));
        pstmt.setDate(7,new
java.sql.Date(appliBonus.getProjectEndDate().getTime()));
        return pstmt.executeUpdate();
    }

//项目删除
    public int appliBonusDelete(Connection con, String projectId) throws
Exception {
        String sql = "delete from t_project where projectId=?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, projectId);
        return pstmt.executeUpdate();
    }

//项目内容更新
    public int appliBonusUpdate(Connection con, AppliBonus appliBonus) throws
Exception {
        String sql = "update t_project set
financialId=?,projectName=?,orderPerformance=?,requestPerformance=?,support
Performance=?,projectStartDate=?,projectEndDate=? where projectId=?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, appliBonus.getFinancialId());
        pstmt.setString(2, appliBonus.getProjectName());
        pstmt.setDouble(3, appliBonus.getOrderPerformance());
        pstmt.setDouble(4, appliBonus.getRequestPerformance());
        pstmt.setDouble(5, appliBonus.getSupportPerformance());
```

```
        pstmt.setDate(6,                                                    new
java.sql.Date(appliBonus.getProjectStartDate().getTime()));
        pstmt.setDate(7,                                                    new
java.sql.Date(appliBonus.getProjectEndDate().getTime()));
        pstmt.setInt(8, appliBonus.getProjectId());
        return pstmt.executeUpdate();
    }
    public boolean haveCenterByProjectName(Connection con, String name) throws
Exception {
        String sql = "select * from t_project t1 where t1.projectName=?";
        PreparedStatement pstmt = con.prepareStatement(sql);
        pstmt.setString(1, name);
        ResultSet rs = pstmt.executeQuery();
        if (rs.next()) {
            return true;
        }
        return false;
    }
}
```

3.项目奖金及各经费额度划分

主要功能：

- 可以对申请过奖金的按总则、细则进行划分
- 可以进行一键划分

图 1-3 显示了主要按钮功能等信息。

图 1-3 划分页面

当按总则细则划分后的数据，可以选择个别申请记录的额度划分进行单独调整具体金额，如图 1-4 所示。

	奖金总额	中心	本部	部门	PM
调整结果	75000	6594	4011	16977	47417
调整额	75000	-500	+100	+400	
按照规则划分	75000	6094	4011	17377	47417

图 1-4 单独调整页面

该模块关键代码如下：

```
public class AppliBonusServlet extends HttpServlet {
    private static final long serialVersionUID = 1L;
    DbUtil dbUtil = new DbUtil();
    AppliBonusDao appliBonusDao = new AppliBonusDao();
    MD5Util md5util = new MD5Util();

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        this.doPost(request, response);
    }

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        request.setCharacterEncoding("utf-8");
        HttpSession session = request.getSession();
        String s_appliBonusText = request.getParameter("s_appliBonusText");
        String searchType = request.getParameter("searchType");
        String page = request.getParameter("page");
        String action = request.getParameter("action");
```

```
AppliBonus appliBonus = new AppliBonus();
if ("preSave".equals(action)) {
    appliBonusPreSave(request, response);
    return;
} else if ("save".equals(action)) {
    appliBonusSave(request, response);
    return;
} else if ("delete".equals(action)) {
    appliBonusDelete(request, response);
    return;
} else if ("list".equals(action)) {
    if (StringUtil.isNotEmpty(s_appliBonusText)) {
        if ("projectName".equals(searchType)) {
            appliBonus.setProjectName(s_appliBonusText);
        }
    }
    session.removeAttribute("s_appliBonusText");
    session.removeAttribute("searchType");
    request.setAttribute("s_appliBonusText", s_appliBonusText);
    request.setAttribute("searchType", searchType);
} else if ("search".equals(action)) {
    if (StringUtil.isNotEmpty(s_appliBonusText)) {
        if ("projectName".equals(searchType)) {
            appliBonus.setProjectName(s_appliBonusText);
        }
        session.setAttribute("searchType", searchType);
        session.setAttribute("s_appliBonusText", s_appliBonusText);
    } else {
        session.removeAttribute("s_appliBonusText");
        session.removeAttribute("searchType");
    }
```

```
        } else {
            if (StringUtil.isNotEmpty(s_appliBonusText)) {
                if ("projectName".equals(searchType)) {
                    appliBonus.setProjectName(s_appliBonusText);
                }
                session.setAttribute("searchType", searchType);
                session.setAttribute("s_appliBonusText", s_appliBonusText);
            }
            if (StringUtil.isEmpty(s_appliBonusText)) {
                Object o1 = session.getAttribute("s_appliBonusText");
                Object o2 = session.getAttribute("searchType");
                if (o1 != null) {
                    if ("projectName".equals((String) o2)) {
                        appliBonus.setProjectName((String) o1);
                    }
                }
            }
        }
        if (StringUtil.isEmpty(page)) {
            page = "1";
        }

        Connection con = null;
        PageBean pageBean = new PageBean(Integer.parseInt(page),
Integer.parseInt(PropertiesUtil.getValue("pageSize")));
        request.setAttribute("pageSize", pageBean.getPageSize());
        request.setAttribute("page", pageBean.getPage());
        try {
            con = dbUtil.getCon();
           List<AppliBonus> appliBonusList = appliBonusDao.appliBonusList(con,
pageBean, appliBonus);
```

```
            int total = appliBonusDao.appliBonusCount(con, appliBonus);
            String pageCode = this.genPagation(total, Integer.parseInt(page),
                    Integer.parseInt(PropertiesUtil.getValue("pageSize")));
            request.setAttribute("pageCode", pageCode);
            request.setAttribute("appliBonusList", appliBonusList);
            request.setAttribute("mainPage", "charger/appliBonus.jsp");
    request.getRequestDispatcher("mainCharger.jsp").forward(request,
response);
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                dbUtil.closeCon(con);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }

    private void appliBonusDelete(HttpServletRequest request,
HttpServletResponse response) {
        String centerId = request.getParameter("projectId");
        Connection con = null;
        try {
            con = dbUtil.getCon();
            appliBonusDao.appliBonusDelete(con, centerId);
    request.getRequestDispatcher("appliBonus?action=list").forward(request,
response);
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
```

```
            dbUtil.closeCon(con);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

private void appliBonusSave(HttpServletRequest request,
HttpServletResponse response)
        throws ServletException, IOException {
    String projectId = request.getParameter("projectId");
    String financialId = request.getParameter("financialId");
    String projectName = request.getParameter("projectName");
    String orderPerformance = request.getParameter("orderPerformance");
    String requestPerformance =
request.getParameter("requestPerformance");
    String supportPerformance =
request.getParameter("supportPerformance");
    Double i1 = Double.parseDouble(orderPerformance);
    Double i2 = Double.parseDouble(requestPerformance);
    Double i3 = Double.parseDouble(supportPerformance);
    String projectStartDate = request.getParameter("projectStartDate");
    String projectEndDate = request.getParameter("projectEndDate");
    DateFormat fmt =new SimpleDateFormat("yyyy-MM-dd");
    Date date1 = null;
    Date date2 = null;
    try {
        date1 = fmt.parse(projectStartDate);
        date2 = fmt.parse(projectEndDate);
    } catch (ParseException e1) {
        e1.printStackTrace();
    }
```

```
        AppliBonus appliBonus = new AppliBonus(financialId, projectName, i1, i2,
i3, date1, date2);
        if (StringUtil.isNotEmpty(projectId)) {
            appliBonus.setProjectId(Integer.parseInt(projectId));
        }
        Connection con = null;
        try {
            con = dbUtil.getCon();
            int saveNum = 0;
            if (StringUtil.isNotEmpty(projectId)) {
                saveNum=appliBonusDao.appliBonusUpdate(con,appliBonus);
            }     else     if     (appliBonusDao.haveCenterByProjectName(con,
appliBonus.getProjectName())) {
                request.setAttribute("appliBonus", appliBonus);
                request.setAttribute("error", "该项目已存在");
                request.setAttribute("mainPage",
"charger/appliBonusSave.jsp");

    request.getRequestDispatcher("mainCharger.jsp").forward(request,
response);
//添加项目的时候进行判定项目是否存在，若存在则弹出提示。

                try {
                    dbUtil.closeCon(con);
                } catch (Exception e) {
                    e.printStackTrace();
                }
                return;
            } else {
```

```
                saveNum = appliBonusDao.appliBonusAdd(con, appliBonus);
            }
            if (saveNum > 0) {

    request.getRequestDispatcher("appliBonus?action=list").forward(request,
response);
            } else {
                request.setAttribute("appliBonus", appliBonus);
                request.setAttribute("error", "保存失败");
                request.setAttribute("mainPage",
"charger/appliBonusSave.jsp");

    request.getRequestDispatcher("mainCharger.jsp").forward(request,
response);
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            try {
                dbUtil.closeCon(con);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
    private void appliBonusPreSave(HttpServletRequest request,
HttpServletResponse response)
            throws ServletException, IOException {
        String projectId = request.getParameter("projectId");
        if (StringUtil.isNotEmpty(projectId)) {
            Connection con = null;
            try {
                con = dbUtil.getCon();
```

```
                AppliBonus appliBonus = appliBonusDao.appliBonusShow(con,
projectId);
                request.setAttribute("appliBonus", appliBonus);
            } catch (Exception e) {
                e.printStackTrace();
            } finally {
                try {
                    dbUtil.closeCon(con);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        }
        request.setAttribute("mainPage", "charger/appliBonusSave.jsp");
        request.getRequestDispatcher("mainCharger.jsp").forward(request,
response);
    }


    private String genPagation(int totalNum, int currentPage, int pageSize) {
        int totalPage = totalNum % pageSize == 0 ? totalNum / pageSize : totalNum
/ pageSize + 1;
        StringBuffer pageCode = new StringBuffer();
        pageCode.append("<a href='appliBonus?page=1'>首页</a>");
        if (currentPage == 1) {
            pageCode.append("<class='disabled'><a href='#'>上一页</a>");
        } else {
            pageCode.append("<a href='appliBonus?page=" + (currentPage - 1) +
"'>上一页</a>");
        }
        for (int i = currentPage - 2; i <= currentPage + 2; i++) {
            if (i < 1 || i > totalPage) {
                continue;
```

```
            }
            if (i == currentPage) {
                pageCode.append("<class='active'><a href='#'>" + i + "</a>");
            } else {
                pageCode.append("<a href='appliBonus?page=" + i + "'>" + i +
"</a>");
            }
        }
        if (currentPage == totalPage) {
            pageCode.append("<class='disabled'><a href='#'>下一页</a>");
        } else {
            pageCode.append("<a href='appliBonus?page=" + (currentPage + 1) +
"'>下一页</a>");
        }
        pageCode.append("<a href='appliBonus?page=" + totalPage + "'>尾 页
</a>");
        return pageCode.toString();
    }
}
```