



南京航空航天大学  
Nanjing University of Aeronautics and Astronautics

# 软件工程综合课程设计

——详细设计

题    目 浙江省政府采购网中标数据采集系统

院    系 计算机技术与科学学院

专    业 软件工程

学生姓名     聂佳，王营    

学    号     161330206    

指导教师     张德平    

二 0 一 五    年    1 2 月    3 1 日

## 目 录

<b>1 引言 .....</b>	<b>3</b>
1.1 编写目的.....	3
1.2 背景.....	3
1.3 基线.....	3
1.4 定义.....	3
1.5 参考资料.....	3
1.6 术语与缩写解释.....	3
<b>2 模块命名规则 .....</b>	<b>4</b>
<b>3 总体设计 .....</b>	<b>4</b>
3.1 需求概述.....	4
3.2 软件描述.....	4
3.3 软件结构.....	4
<b>4 模块汇总 .....</b>	<b>6</b>
4.1 模块汇总表.....	7
4.2 模块关系图.....	7
<b>5 子系统模块设计 .....</b>	<b>7</b>
5.1 模块 1.....	7
5.2 模块 2.....	8
5.3 模块 3.....	9
5.4 模块 4.....	11
<b>6 界面详细设计 .....</b>	<b>11</b>
6.1 界面外部设计.....	12
6.2 界面内部设计 .....	13
6.3 用户界面设计 .....	14
<b>7 信息编码设计 .....</b>	<b>14</b>
7.1 代码结构设计 .....	14
7.2 代码编制 .....	14

# 1 引言

## 1.1 编写目的

本说明书在概要设计的基础上，对政府采购网中标数据采集系统的各模块、查询、子系统分别给予了不同层面上的要求，目的在于明确说明系统各功能的实现方式，指导开发员进行编码。

本说明书的预期读者为：模块开发人员、后期维护人员等。

## 1.2 背景

待开发软件系统的名称：政府采购网中标数据采集系统

此软件系统任务提出者：软件工程综合课设

此软件系统任务开发者：胡思旺、王营、王宏、聂佳

## 1.3 基线

政府采购网中标数据采集系统需求分析说明书 v1.0

## 1.4 定义

- 政府采购网中标数据采集系统：该系统是基于 JAVA 及爬虫技术的一个数据采集系统。

## 1.5 参考资料

属于本项目的其他已发表的文件。

本文件中引用的其他文献、资料以及软件开发标准。

## 1.6 术语与缩写解释

缩写、术语	解 释
SQL server	数据库
SD	系统设计，System Design
B/S	Browser/Server
网络爬虫	一种按照一定的规则，自动地抓取万维网信息的程序或者脚本

## 2 模块命名规则

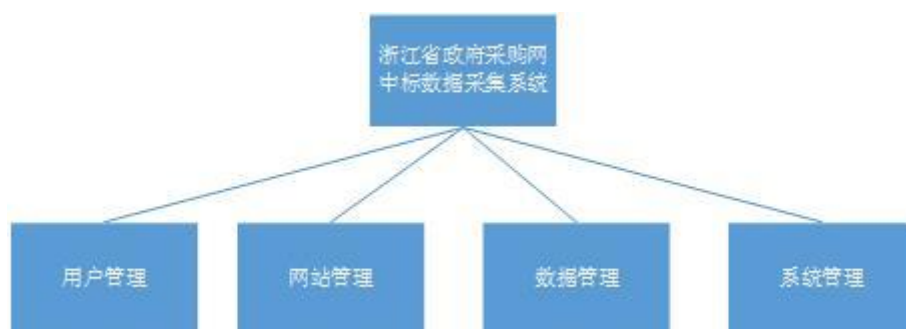
## 3 总体设计

### 3.1 需求概述

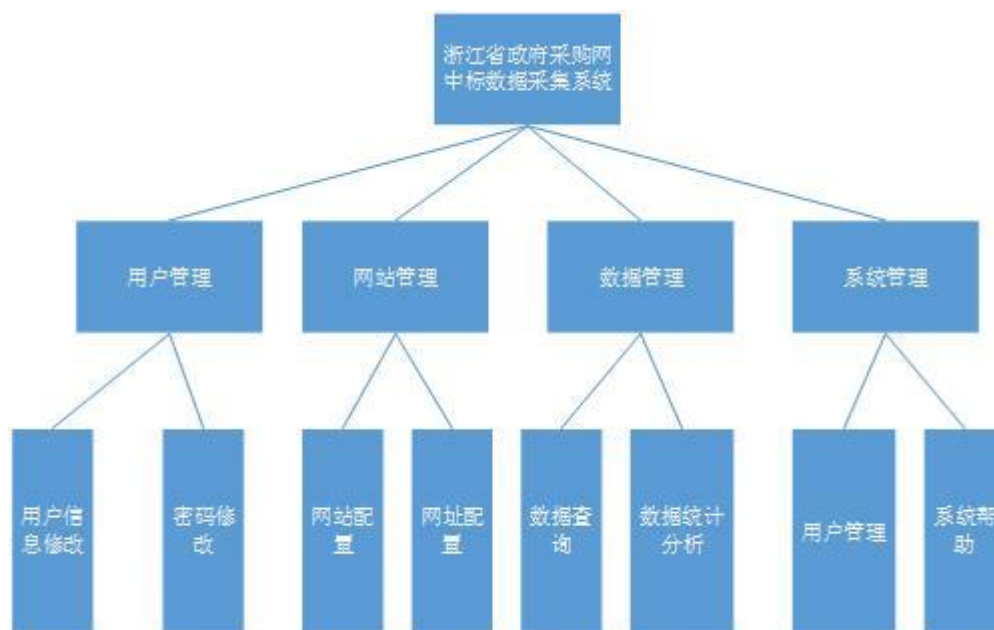
随着大数据分析技术的兴起，数据成为大数据分析的重要基础，大数据的采集已成为直接影响大数据分析技术应用的瓶颈，政府采购网中标数据采集工具可用于对各省、市级政府采购网中标数据的采集，为企业景气指数分析、企业生产状况分析、企业融资风险分析以及股民股市投资提供分析数据，并在此基础上提取相应的辅助决策信息。

### 3.2 软件描述

### 3.3 软件结构



系统结构图



系统层次结构图

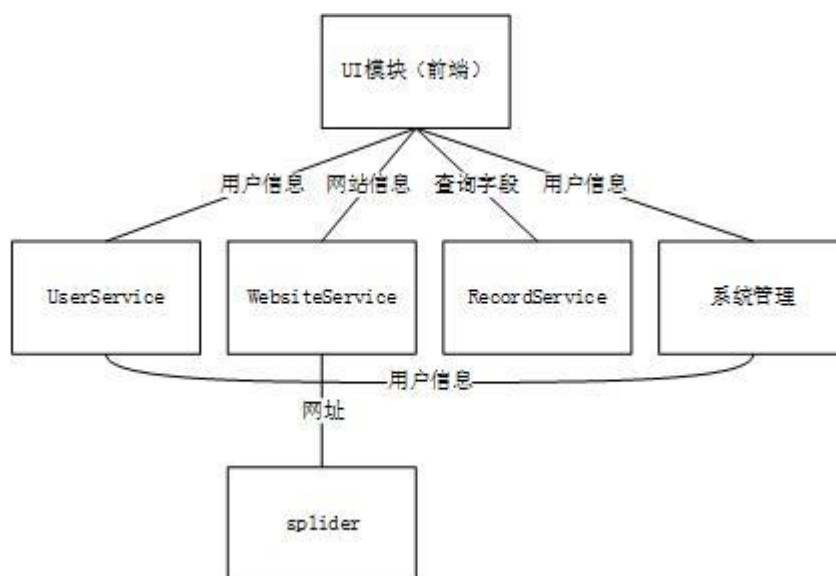
## 4 模块汇总

### 4.1 模块汇总表

子系统 用户信息管理	
模块名称	功能简述
UserService	对用户实体类进行基本的操作，如增删改查
子系统 网站管理	
模块名称	功能简述
WebService	对 website 实体类进行操作，达到配置网站信息，配置网址等目的
Splicer	配置好需要收集数据的网站后，对网站进行爬虫获得数据
子系统 数据管理	
模块名称	功能简述

RcordService	根据用户选择的查询字段，对数据库中的数据进行查询，并将结果进行数据分析
子系统 系统管理	
模块名称	功能简述

## 4.2 模块关系图



# 5 子系统模块设计

## 5.1 模块 1

模块名称	<i>UserService</i>
功能描述	登录、注册、用户信息修改、密码修改
接口与属性	
数据结构与算法	该模块根据传入的用户类的对象，或该对象的某一成员变量，调用数据库层对应的基本操作方法，对数据库中 User 的信息进行增删改查  <pre> private UserDao userDao;  public UserService() {     userDao = new UserDao(); }                     </pre>



```
public User find(String uuid) throws SQLException{
    return userDao.find(uuid);
}

public boolean register(User user) throws SQLException
{
    if (userDao.findUser(user.getAccount())) {
        throw new RuntimeException("该用户不存在");
    } else {
        userDao.addUser(user);
        return true;
    }
}

public User login(String account, String password)
throws SQLException {
    if (userDao.findUser(account, password) != null) {
        return userDao.findUser(account, password);
    } else {
        throw new RuntimeException("密码错误");
    }
}

public void delete(User user) throws SQLException{
    userDao.deleteUser(user);
}

public List<User> getAll() throws Exception{
    return userDao.getAll();
}

public void update(User user) throws SQLException{
    userDao.updateUser(user);
}
}
```

数据结构：自定义 User 类

## 5.2 模块 2

模块名称	WebsiteService
功能描述	管理网站信息，配置网址



接口与属性	
数据结构 与算法	<p>该模块根据传入的 Website 类的对象，或该对象的某一成员变量，调用数据库层对应的基本操作方法，对数据库中的 Website 的信息进行增删改查</p> <pre>public class WebSiteService {     private WebSiteDao webSiteDao;      public WebSiteService() {         webSiteDao = new WebSiteDao();     }      public WebSite find(String uuid) throws SQLException{         return webSiteDao.find(uuid);     }      public void add(WebSite webSite) throws SQLException{         webSiteDao.addWebSite(webSite);     }      public void delete(WebSite webSite) throws SQLException{         webSiteDao.deleteWebSite(webSite);     }      public void update(WebSite webSite) throws SQLException{         webSiteDao.updateWebSite(webSite);     }      public List&lt;WebSite&gt; getAll() throws SQLException{         return webSiteDao.getAll();     } }</pre> <p>数据结构：自定义的 Website 类</p>

## 5.3 模块 3

模块名称	<i>Splider</i>
功能描述	对相应的网址进行爬虫，抓取数据
接口与属性	方法：Splider，入口参数：String url（网址）
数据结构 与算法	数据结构：Record <pre>public class Splider {</pre>





```
public void splider(String url) throws
FailingHttpStatusCodeException, MalformedURLException,
IOException, InterruptedException, SQLException{
    Example example=new Example();
    example.splider(url);
}

public class Example {

    public static void main(String[] params)
        throws FailingHttpStatusCodeException,
        MalformedURLException,
        IOException, InterruptedException,
        SQLException {
        LoggerFactory.getFactory().setAttribute(
            "org.apache.commons.logging.Log",
            "org.apache.commons.logging.impl.NoOpLog");

        java.util.logging.Logger.getLogger("com.gargoylesoft
ware.htmlunit")
            .setLevel(Level.OFF);

        java.util.logging.Logger.getLogger("org.apache.commo
ns.httpClient")
            .setLevel(Level.OFF);

        String ningbo =
"http://www.nbzfcg.cn/project/MoreNotice.aspx?Type=3";
        String hangzhou =
"http://cg.hzft.gov.cn/www/noticelist.do?noticetype=5,6
";
        String
lishui="http://www.lscgzb.gov.cn/index.php?m=content&c=
index&a=lists&catid=20";
        new Example().splider(ningbo);
    }

    public void splider(String url)
        throws FailingHttpStatusCodeException,
        MalformedURLException,
        IOException, InterruptedException,
        SQLException {
        url=url.replaceAll(" ", "");
    }
}
```



	<pre>        if(url.equals("http://www.nbzfcg.cn/project/MoreNotice.aspx?Type=3")){             Ningbo.spliderwork();         }          if(url.equals("http://cg.hzft.gov.cn/www/noticelist.do?noticetype=5,6")){             Hangzhou.spliderwork();          }if(url.equals("http://www.lscgzb.gov.cn/index.php?m=content&amp;c=index&amp;a=lists&amp;catid=20")){             Lishui.splider();          }if(url.equals("http://www.zszfcg.gov.cn/news/96.html")){             Zhongshang.splider();         }else{             Ningbo.spliderwork();         }     } }</pre>
--	---

## 5.4 模块 4

模块名称	<i>RecordService</i>
功能描述	根据用户选择的查询字段，查询数据库中的数据，并将所得的结果进行数据分析，用图表的方式显示
接口与属性	
数据结构与算法	<p>该模块根据传入的查询字段，调用数据库层的对应的基本查询方法，查询数据库中的数据，将查询到的记录进行统计分析，向调用模块返回图表的相关元素</p> <pre>public class RecordService {     RecordDao recordDao;      public RecordService() {         recordDao = new RecordDao();     }      public List&lt;Record&gt; find(String keyword) throws SQLException {         return recordDao.find(keyword);     } }</pre>



```
public List<Record> findById(int id) throws
SQLException{
    return recordDao.findById(id);
}

public List<Record> findByName(String name) throws
SQLException {
    return recordDao.findByName(name);
}

public List<Record> findbyUnit(String unit) throws
SQLException {
    return recordDao.findbyUnit(unit);
}

public List<Record> findbyTime(String time) throws
SQLException {
    return recordDao.findbyTime(time);
}

public List<Record> findbyMoney(String money) throws
SQLException {
    return recordDao.findbyMoney(money);
}

public List<Record> getPage(int start, int count)
throws SQLException {
    return recordDao.getPage(start, count);
}

public List<Record> getAll() throws SQLException {
    return recordDao.getAll();
}

public void exportRecord(String fileName) throws
SQLException {
    recordDao.exportRecord(fileName);
}
}
```

数据结构: Record



## 5.4 系统界面详细设计

### 5.4.1 外部界面设计

本系统的主界面：

```
public class Home extends HttpServlet {  
  
    public void doGet(HttpServletRequest request, HttpServletResponse  
response)  
        throws ServletException, IOException {  
        request.getRequestDispatcher("/View/home.jsp").forward(request,  
response);  
    }  
  
    public void doPost(HttpServletRequest request, HttpServletResponse  
response)  
        throws ServletException, IOException {  
        doGet(request, response);  
    }  
  
}
```

登录界面：

```
public class Login extends HttpServlet {  
  
    public void doGet(HttpServletRequest request, HttpServletResponse  
response)  
        throws ServletException, IOException {  
  
        request.getRequestDispatcher("/View/login.jsp").forward(request,  
response);  
    }  
  
    public void doPost(HttpServletRequest request, HttpServletResponse  
response)  
        throws ServletException, IOException {  
        doGet(request, response);  
    }  
  
}
```



### 5.4.1 内部界面设计

### 5.4.2 用户界面设计

本系统的用户界面设计采用扁平化风格，左边是菜单导航栏，右边是显示内容的窗口。用户信息的输入均采用表单方式提交。

## 5.信息编码设计

### 5.5 代码结构设计

*确认信息分类编码总体方案，进行分类代码结构设计。*

### 5.6 代码编制

*按代码结构编制信息代码*