解决方案说明

Solution Model Description

**编号**：**TMP-SMD**

**版本 1.0**

|  |  |  |  |
| --- | --- | --- | --- |
| 作者： |  | 日期： |  |
| 审批： |  | 日期： |  |

**变更记录**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 变更说明 | 作者 |
|  | 1.0 | 创建 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[1 引言 1](#_Toc282072460)

[2 系统整体架构 2](#_Toc282072461)

[3 类设计 4](#_Toc282072462)

[4 模型层(Model)解决方案 5](#_Toc282072463)

[3.1业务层解决方案 5](#_Toc282072464)

[3.2持久层解决方案 5](#_Toc282072465)

[5 表示层(View)解决方案 7](#_Toc282072466)

[6 控制器（Servlet、Action类） 8](#_Toc282072467)

[7 其他类（Model类） 9](#_Toc282072468)

# 引言

解决方案（Solution Model）是在鲁棒分析的设计模型基础上，进一步完成到计算机系统的设计。

# 系统整体架构

采用Spring MVC(Model,View,Controller)应用框架实现。

Browser

Response

Request

View(JSP)

Controller(Servlet)

Spring MVC

业务逻辑层(Action)

业务层(Service)

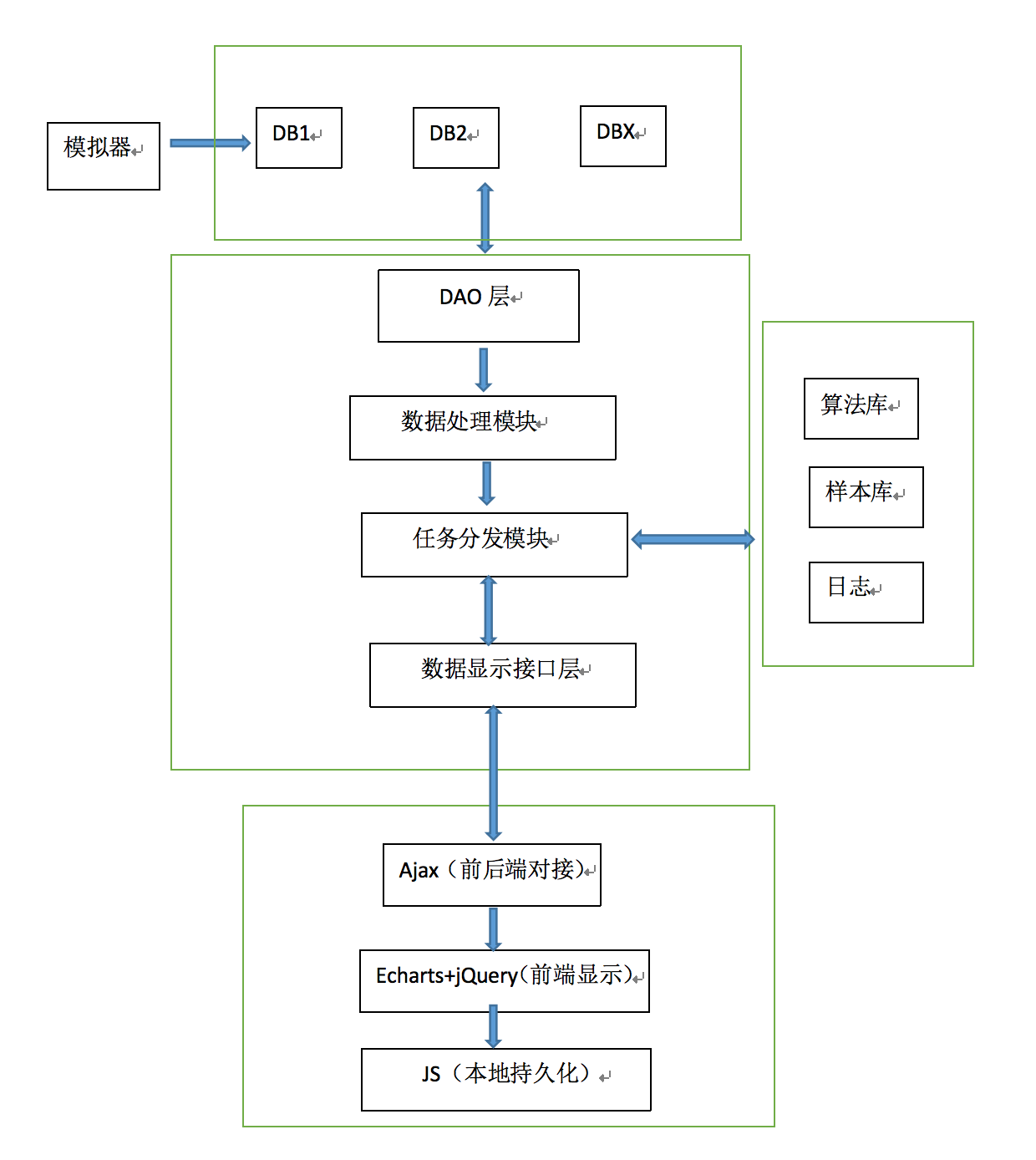
DAO层

Model层c

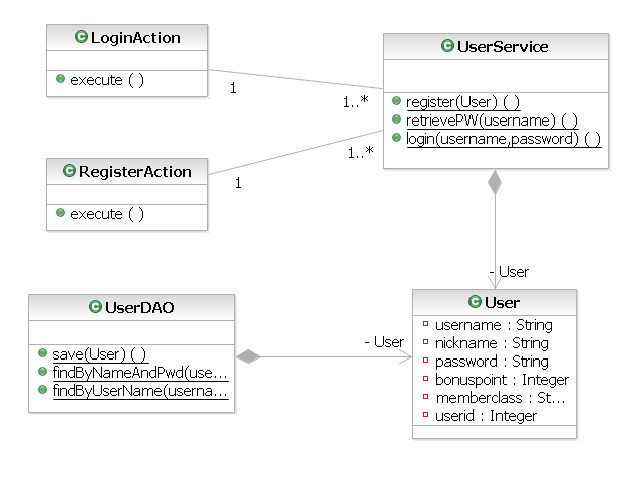
Mybatis持久化对象

数据层（DB）

模块表示：



# 类设计



# 模型层(Model)解决方案

使用Mybatis应用框架实现。模型层又分为业务层和持久层。业务层处理Spring MVC的Action发过来的业务逻辑请求，持久层与数据库的操作。业务层不直接操作数据库，通过调用持久层来间接操作数据库。

### 3.1业务层解决方案

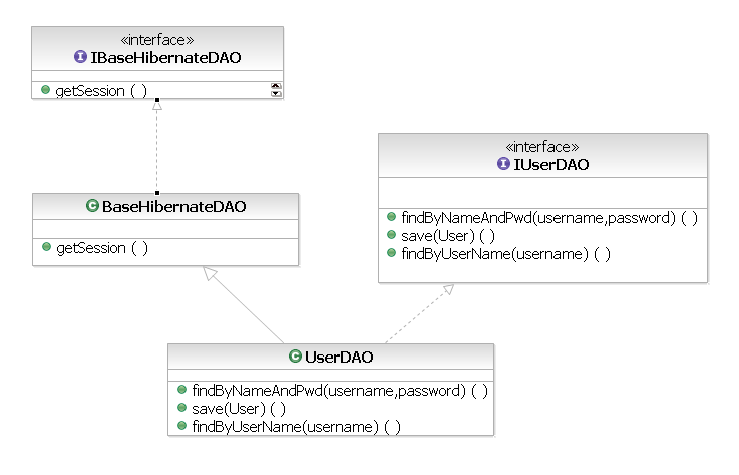
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 类名称 | | UserService | | | 所属包名称 | com.aiop.users | | |
| 类静态变量定义 | | | | | | | | |
| 名称 | | | 类型 | | | | 描述 | |
|  | | |  | | | |  | |
| 类变量定义(先Public->protected->private) | | | | | | | | |
| 名称 | | | 类型 | | | | 描述 | |
|  | | |  | | | |  | |
|  | | |  | | | |  | |
| 方法(先Public->protected->private) | | | | | | | | |
| 名称 | 参数 | | | 返回值 | | | | 简单描述 |
| Login | String username,String password | | | static boolean | | | | 判定管理员登陆信息是否有效 |
| register | Users user | | | static boolean | | | | 注册管理员 |
| retrievePW | String username | | | static boolean | | | | 找回密码功能 |
| inspectSys | void | | | void | | | | 发送监控请求 |

### 3.2持久层解决方案

持久层组件是完成数据库处理的组件，例如数据源的管理、执行数据库的查询、CRUD。

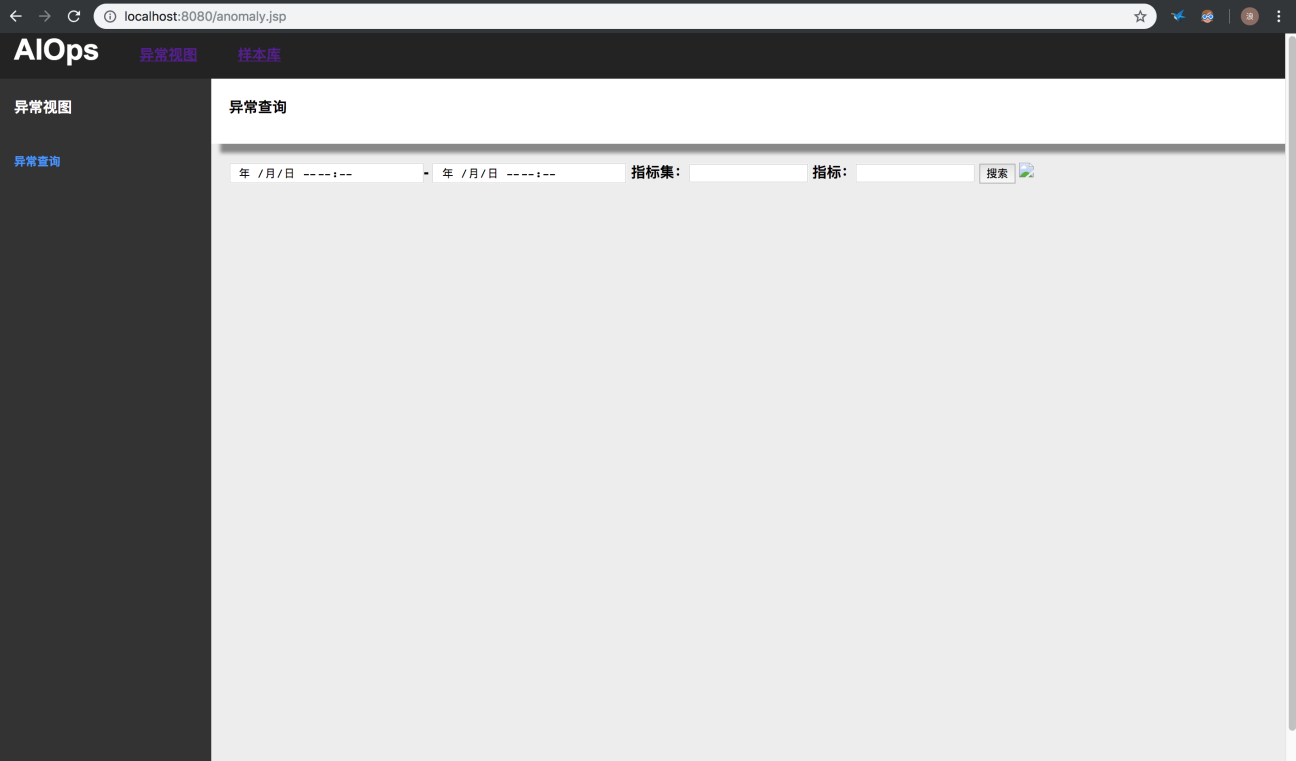
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 类名称 | | UserDAO | | | 所属包名称 | com.aiop.users | | |
| 类静态变量定义 | | | | | | | | |
| 名称 | | | 类型 | | | | 描述 | |
|  | | |  | | | |  | |
| 类变量定义(先Public->protected->private) | | | | | | | | |
| 名称 | | | 类型 | | | | 描述 | |
|  | | |  | | | |  | |
|  | | |  | | | |  | |
| 方法(先Public->protected->private) | | | | | | | | |
| 名称 | 参数 | | | 返回值 | | | | 简单描述 |
| findByNameAndPwd | String username,String password | | | List | | | | 核对用户名和密码是否有效 |
| Save | Users | | | void | | | | 向数据库中插入一条新会员数据记录 |
| getNewData | int timestamp | | | List | | | | 根据时间戳获取相应的时序数据 |

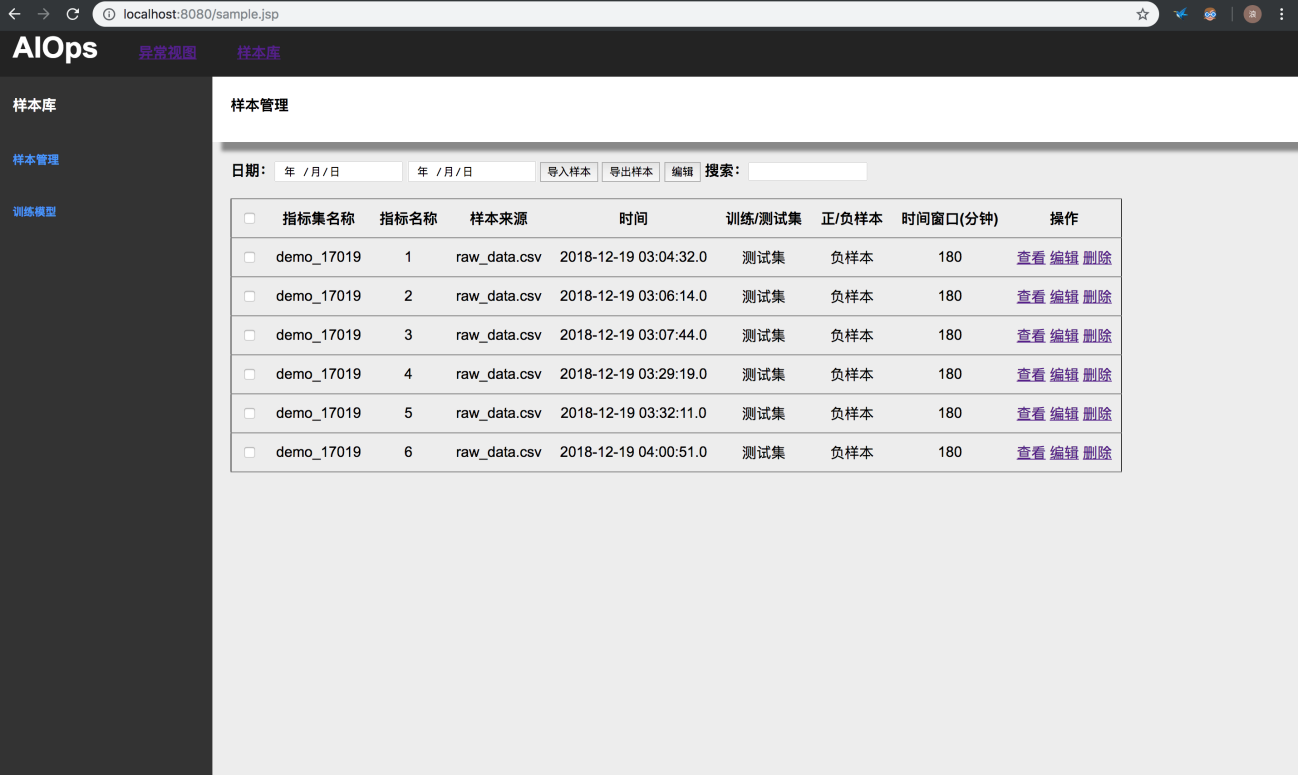
下图为Hibernate应用框架实现的DAO层结构



# 表示层(View)解决方案

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **JSP页面说明** | | | | | | |
| 页面名称 | | login.jsp、anomaly.jsp、sample.jsp | | | | |
| 功能简述 | | 管理用户登录；  显示系统实时异常监控信息；  显示样本管理页面。 | | | | |
| 进入的URL | | /login.jsp | | | 进入方式 | dispatcherServlet调度 |
| 数据来源定义 | | | | | | |
| 编号 | 参数名称 | | 来源 | 说明 | | |
| 1 |  | |  |  | | |
| 页面外观 | | | | | | |
|  | | | | | | |





# 控制器（Servlet、Action类）

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Servlet类说明** | | | | | | |
| 类名称 | | | com.aiop.vertify、com.aiop.getlist、com.aiop.detect、com.aiop.dispatch | | | |
| 功能简述 | | | 调用算法进行时序数据检测  执行的操作如下：   1. vertify验证管理员信息 2. getlist从接入的数据库中获取实时数据 3. detect调用算法对实时数据进行异常检测 4. diapatch将实时检测结果分发给前端页面 | | | |
| 输入定义 | | | | | | |
| # | | 名称 | | 类型 | | 说明 |
| 1 | | Timestamp | | long | | 时序数据时间戳，从实时系统中按时刻获取 |
| 处理流程 | | | | | | |
| 取出该会员的会员信息以及行业信息，传递给后面的jsp | | | | | | |
| 输出定义 | | | | | | |
| # | 名称 | | | 类型 | 说明 | |
| 1 | DataFrame | | | list | 将时序数据从数据库中取到DataInfo数据对象中，通过该对象存储的数据进行异常检测，将检测结果存储到DataResult中，并将该DataInfo对象及DataResult存放到Request中，传递给下面的JSP页面显示 | |

# 其他类（Model）

主要包括：DataInf

public class DataInf {  
 private String kpiId;  
 private long timestamp;  
 private float value;  
  
 public DataInf() {}  
  
 public String getKpiId() {return kpiId;}  
  
 public void setKpiId(String kpiId) {this.kpiId = kpiId;}  
  
 public long getTimestamp() {return timestamp;}  
  
 public void setTimestamp(long timestamp) {this.timestamp = timestamp;}  
  
 public float getValue() {return value;}  
  
 public void setValue(float value) {this.value = value;}  
  
 public DataInf(String kpiId, long timestamp, float value) {  
 this.kpiId = kpiId;  
 this.timestamp = timestamp;  
 this.value = value;  
 }  
}