3.1 系统结构

Spring MVC框架是Spring提供的一个构建Web应用程序的MVC实现，通过实现Model-View-Controller模式很好的将数据对象、业务逻辑和表现展示进行分离，降低系统的耦合性如图3-1所示。

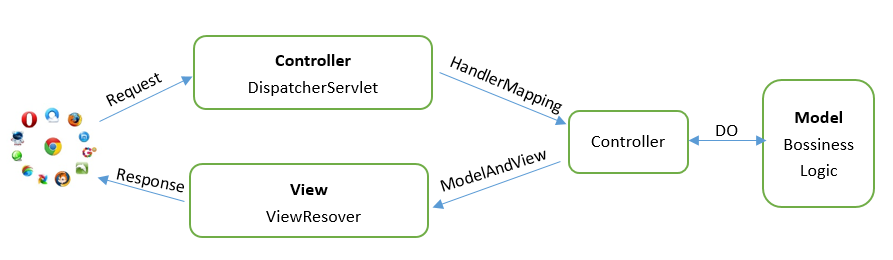


图3-1.Spring MVC的结构

当浏览器发起request，Spring MVC的核心控制器DispatcherServlet接收到请求，对请求做出相应的预处理，借助HandlerMapping返回一个适当的处理程序，同样该程序也是一个Controller，再在该Controller中完成业务逻辑，并将返回的对象ModelAndView对象传递给ViewResolver，通过读取ViewResolver的配置，将视图对象展示给用户。

本系统使用Spring MVC框架，将订水管理系统分为表现层、控制层、业务逻辑层、持久层四层，如图3-2所示

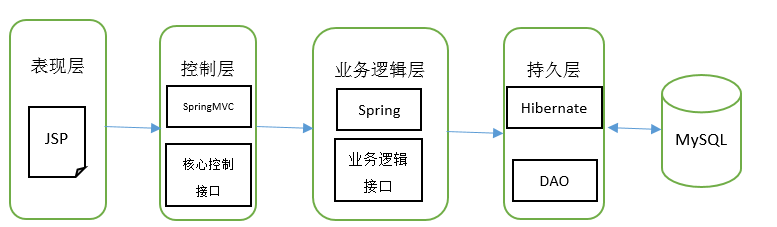


图3-2.基于Spring MVC的系统分层模型

3.1.1 视图与控制器之间的接口

在众多的视图技术中，最常用的是JSP，此外Spring MVC还支持Velocity、Tiles、iText 和POI等。为了能够将控制器传递来的Model在表现层展示出来，需要对其进行相关的配置——applicationContext.xml（当然Spring MVC允许可以自定义配置文件的名字，而这需要在web.xml中加以配置），具体配置如图3-3，



图3-3.Spring MVC加载配置文件的配置

在dispatcher-servlet.xml中使用bean 配置的方式添加表现层视图解析器的配置，如图3-4所示，

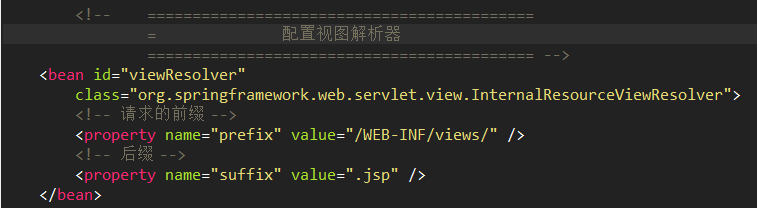


图3-4.视图解析器的配置

通过将控制器返回的URL的后缀匹配的方式，将会在/WEB-INF/views/文件夹下找到相应的JSP文件，如上配置，后缀是不用匹配的，默认为JSP。由于spring框架是一个轻量级，无入侵性的框架实现方式，MVC实现的可扩展性很强、可以很容易地将你选择的Web框架和Spring结合起来[引。只要通过Spring的ContextLoadListener启动一个Spring的根应用上下文，并且通过它的ServletContext属性在Struts或WebWork的Action中访问[3-1]。

3.2 服务器端设计

3.2.1 系统功能结构图

系统主要功能结构如图3-5所示，

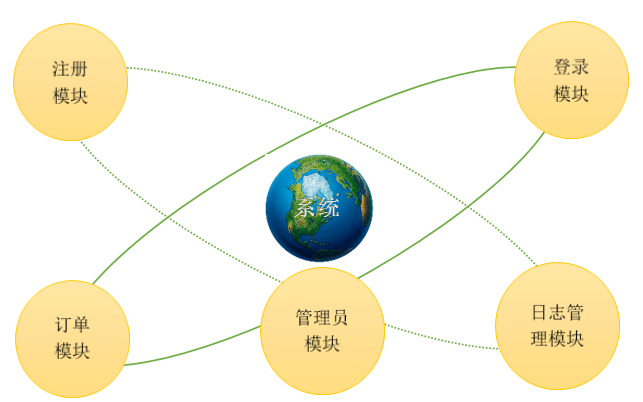


图3-5.系统功能结构

在用户发起登录/注册服务的时候，使用Hibernate Validator和JQuery方式完成对表单的数据验证，服务器使用MySQL作为存储介质，并通过Hibernate ORM完成对数据CURD操作。服务器端完成的任务是提供浏览器访问接口和实现相关业务逻辑处理。服务器端的业务逻辑的包组织结构如图3-6所示：

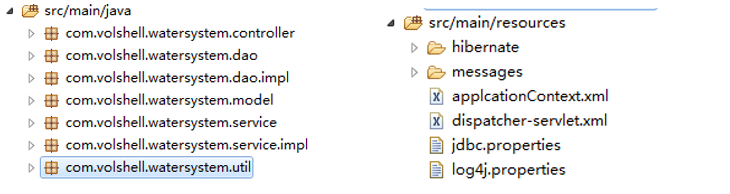


图3-6.系统包组织结构

各个包的功能如下：

1. com.volshell.watersystem.controller 系统核心控制器
2. com.volshell.watersystem.dao 系统数据访问对象接口
3. com.volshell.watersystem.dao.impl 数据访问对象的实现类
4. com.volshell.watersystem.model 系统业务模型
5. com.volshell.watersystem.service 系统提供的服务接口
6. com.volshell.watersystem.service.impl 服务接口实现类
7. com.volshell.watersystem.util 整个系统的工具类

3.2.2 系统的组织管理

本系统是用maven构建，并对相应依赖包的管理，本系统中所依赖的包，如下图3-7所示。同时使用GitHub对系统代码进行托管，托管地址为<https://github.com/smartvolshell/WaterSystem>。

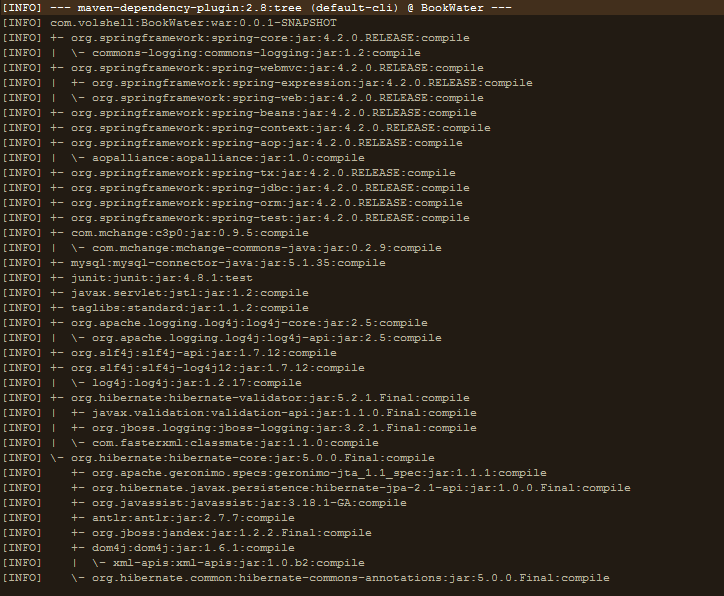


图3-7.系统依赖包结构

3.2.3 业务逻辑层的设计和实现

在系统逻辑结构中，业务逻辑层位于表现层之下，数据持久层之上，依赖特定的业务规则来实现业务的处理，通过向控制层开放接口来实现相关服务，同时调用持久层的数据处理逻辑完成数据的持久化操作。综上可用下图3-8来表示，

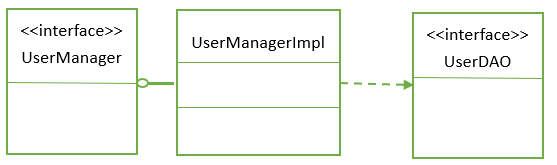


图3-8. 业务逻辑层组件之间的关系

3.2.4 数据持久层的设计和实现

数据持久层通过DAO向业务逻辑层提供服务接口，DAO的实现包括配置数据源（本系统使用C3P0数据源配置）、事务管理，构建BaseDAO基类，在基类的实现类中使用Hibernate @Entity注解实现ORM，实现了常用的CURD等数据访问操作。本系统中数据持久层之间的关系如下图3-9所示

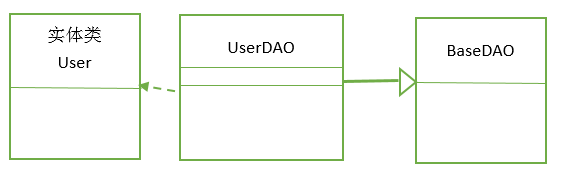


图3-9. 数据持久层组件之间的关系

[3-1]. Totobo. 使用Struts + Spring + Hibernate组装Web应用[EB/OL]

<http://www.kissjava.com/doc/J2ee/websevice/2005-06-07/19451118128953.html> .2013.