成 绩:\_\_\_\_\_\_\_\_\_\_\_\_

教师签名:\_\_\_\_\_\_\_\_\_\_\_\_



**需求分析说明书**

**项目名称:** 超市账单管理系统

教 师: 胡峡进

姓 名: 唐可寅

学 号: 201710253110

班 级: 17gb计算3班

日 期: 2020年07月06日

目录

[一、概述 1](#_Toc538864094)

[1、摘要 1](#_Toc750288459)

[2、系统的应用背景 1](#_Toc74155229)

[3、系统的开发意义 1](#_Toc786418543)

[二、需求分析 2](#_Toc1722088563)

[1、可行性研究 2](#_Toc1505367722)

[三、系统总体设计 3](#_Toc1210458347)

[1、系统设计 3](#_Toc1060849998)

[系统功能分析简述 3](#_Toc1296678992)

[项目环境和依赖 3](#_Toc619768788)

[分层分析 4](#_Toc1158331966)

[2、系统详细设计 10](#_Toc1146092507)

[前言 10](#_Toc1595935206)

[概要设计 10](#_Toc812256212)

[详细设计 11](#_Toc36611105)

[四、系统的编码、测试和总结 17](#_Toc1142518693)

[1、系统代码 17](#_Toc1660385424)

[2、系统的测试 19](#_Toc1695312050)

[五、总结 21](#_Toc296595954)

# 一、概述

## 1、摘要

随着信息化技术在生活中的应用越来越广泛，网络即时聊天也逐渐成为人们的一种生活方式。在线聊天室系统室在此环境下应运而生。聊天室是一项进行文本交互的软件，应用广泛，并且实用性强的一个非常典型的人机交互系统。在信息世界里，聊天工具的使用是非常频繁的，如 MSN、QQ、校内通之类的聊天工具许多人都正在使用。

本次设计首先对系统建设对各个层面进行了基本定义，并在其基础上进行了简单分析，阐述中给出了部分UML图以说明系统结构。本次设计以基于文本信息传送和接收功能的聊天室系统为基准实行此次系统的开发。

## 2、系统的应用背景

聊天室作为即时通讯的一种，即时通讯（Instant Messenger，简称 IM）软件可以说是目前我国上网用户使用率最高的软件，无论是老牌的 ICQ，还是国内用户量第一的腾讯QQ, 以及微软的 MSN Messenger 都是大众关注的焦点，它们能让你迅速的在网上找到你的朋友或工作伙伴，可以实时交谈和互传信息。而且，现在不少 IM 软件还集成了数据交换、语音聊天、网络会议、电子邮件的功能。

IM 软件的历史并不久远，但是它一诞生，就立即收到网民的喜爱，并风靡

全球。在它的发展史上，以色列人是功不可没的。正是四位以色列籍的年轻人，在 1996 年 7 月成立的 Mirabilis公司，并于同年 11 月推出了全世界第一个即时通讯软件 ICQ，取意为“我在找你”——“I Seek You”，简称ICQ。直到现在，ICQ 已经推出了它的 2020 版本，在全球即时通讯市场上占有非常重要的地位。

目前，国内最为流行的即时通讯软件是 OlCQ（简称 QQ）。它以良好的中文界面和不断增强的功能形成了一定的 QQ 网络文化。Messenger 虽出道较晚，但依托微软的强大背景，实力也不可小视。最为 Windows XP 的一部分，Messenger 整合了操作系统的许多功能，如多种形式的聊天选择、多人的单窗口讨论式交流以及充分的文件与桌面共享功能等。它令人耳目一新的中文界面和注册方式，连同它强大的功能了，着实吸引了众多的眼球的注视，此外还有许多有特点的 IM 软件。

本次设计将针对网络聊天的特点，开发一套可以用于局域网环境的Web即时聊天系统。

## 3、系统的开发意义

网络聊天系统可以使人在很远的距离内进行聊天，甚至包括语音和视频的。通过网络聊天系统，在可以即时聊天的同时，还可以增进人们之间的感情，使这个世界越来越小，越来越方便。除此之外，它还越来越多地被作为办公系统来使用。如此多的优点之下开发个简单聊天系统是很有意义的。

# 二、需求分析

## 1、可行性研究

（1）技术可行性

本聊天系统采用了Java语言开发，Java语言是 Internet上最热门的编程语言。

Java是一种简单的、面向对象的、分布式的、解释型的、健壮安全的、结构中立的、可移植的、性能优异、多线程的动态语言。在 Java 出现以前，Internet上的信息内容都是一些乏味死板的HTML文档。这对于那些迷恋于WEB浏览的人们来说简直不可容忍。他们迫切希望能在WEB中看到一些交互式的内容，开发人员也极希望能够在WEB上创建类无需考虑软硬件平台就可以执行的应用程序，当然这些程序还要有极大的安全保障。对于用户的这种要求，传统的编程语言显得无能为力。SUN的工程师敏锐地察觉到了这一点，从1994 年起，他们开始将OAK技术应用于WEB上，并且开发出了HotJava的第一个版本。当SUN 公司 1995 年正式以Java这个名字推出的时候，几乎所有的 WEB 开发人员都心生感叹：噢，这正是我想要的！于是Java成了一颗耀眼的明星，丑小鸭一下了变成了白天鹅。

Java以其强安全性、平台无关性、硬件结构无关性、语言简洁同时面向对象，在网络编程语言中占据无可比拟的优势，成为实现网上聊天系统的首选语言。

1. 社会可行性

网络聊天系统为人们提供了一个即时交流和联系的平台。利用现代先进的互联网资源和技术优势，通过完善的网络即时聊天管理信息系统，达到增进人与人之间的信息交互和沟通的目的，并以此为基础加快社会的信息化建设。

近年来计算机技术的快速发展，特别是计算机网络的发展，越来越深刻的改变了人们生活的方方面面，使得人们能以更低廉的价格，开发出更方便、更实用的网络工具。各种在线服务系统，更是深刻的影响了人们的联系方式使得人们可以在远隔千里之遥随时通讯。网络聊天系统作为一种方便人们之间联系的实用系统便应运而生。网络聊天系统是为人与人之间进行交流和联系提供的一个平台。通过提供完善的聊天服务和规范的管理，可以达到增进人与人之间的感情，促进人们之间的沟通，方便人们联系的目的：同时本系统可以拉近人与人之间的距离，丰富人们的业余时间，为人们在网络中交友提供了便捷的渠道。

目前已有很多成功开发聊天软件的先例，社会需要优秀的网上聊天系统为人们提供交流的平台。聊天软件开发和运行与国家的政策法规不存在任何冲突和抵触之处。

## 2、系统需求分析

此次开发的网络聊天工具可作为个人的交流工具使用，通信的安全性不是很高，但要求信息的响应速度要较快让用户充分享受到网络即时消息的方便和快捷。本聊天系统由服务器端程序和浏览器端页面两部分组成，整体采用 JAVA 平台开发，并使用MySQL数据库管理数据。服务器是可以监听和查看用户的基本操作；浏览器页面面向实际用户，它有必要的界面的按钮，向用户提供网络即时消息的功能，本聊天需包含如下基本功能：

1. 浏览器端页面
   1. 提供用户登录、注册退出、访问控制的能力；
   2. 提供浏览在线用户列表的能力；
   3. 提供消息发送能力；
   4. 提供消息接收能力；
   5. 提供良好的用户界面；
2. 服务端
   1. 提供屏蔽用户不文明用语的能力；
   2. 向所有在线用户提供消息转发能力。

# 三、系统总体设计

## 1、系统设计

### 系统功能分析简述

聊天室是一项进行文本交互的软件，应用广泛，并且实用性强的一个非常典型的人机交互系统。在信息世界里，聊天工具的使用是非常频繁的，如 MSN、QQ、校内通之类的聊天工具许多人都正在使用。此次设计的聊天室系统主要针对文本信息的传送和接收，不能像QQ一样进行图片、音频、视频等文件的传输。虽然其功能简单，但是在开发技术和过程中，能够充分学习和使用Java的技术，进一步培养了面向对象程序设计的思想，加深了我们对面向对象程序设计四大特性的理解。

人们日常生活中越来越多地使用聊天室这项应用来为自己的工作和学习服务。一个操作简单，界面友好、运行稳定的聊天室对于小型局域网的用户可以起到很好的交流作用。所谓聊天室就是一个可以让许多用户同时互相通信的网上园地，个人用户可以通过聊天室将自己融入整个主流社会中。

根据目前使用聊天室的用户需求，本次课程设计将完成以下功能需求：

1. 提供用户登录和注册、退出的功能，系统不允许匿名访问；
2. 实现用户按下发送键，消息实时传递给其他用户，当前用户回显此消息，消息为空时不允许发送；
3. 用户在线情况下，长时间（3分钟）未操作即自动退出登录，用户界面返回用户登录页面；
4. 实现当用户名重复时不允许登录，并显示提示信息；
5. 实时统计在线用户数，用户退出或长时间无操作时自动核减在线用户数；
6. 实现不文明内容的屏蔽；
7. 实现对聊天记录的存储功能；
8. 提供友好的用户界面。

### **项目环境和依赖**

* 开发环境
  + 开发语言：JDK 1.8(1.8.0\_221)
  + 开发工具：IntelliJ IDEA 2020.1
  + 应用服务器：Tomcat 7(7.0.96)
  + 数据库：MySQL 8.0.20
* 环境配置
  + 访问路径：http://127.0.0.1:8080/chatRoom/
  + 服务端口号：8080
* 第三方库依赖
  + commons-beanutils 1.9.4
  + commons-collections 3.1
  + commons-lang 2.6
  + commons-logging 1.1
  + ezmorph 1.0.6
  + fastjson 1.2.68
  + javax.servlet-api 3.1.0
  + Json-lib-2.4-jdk15
  + mysql-connector-java 8.0.20
  + jQuery 1.12.4

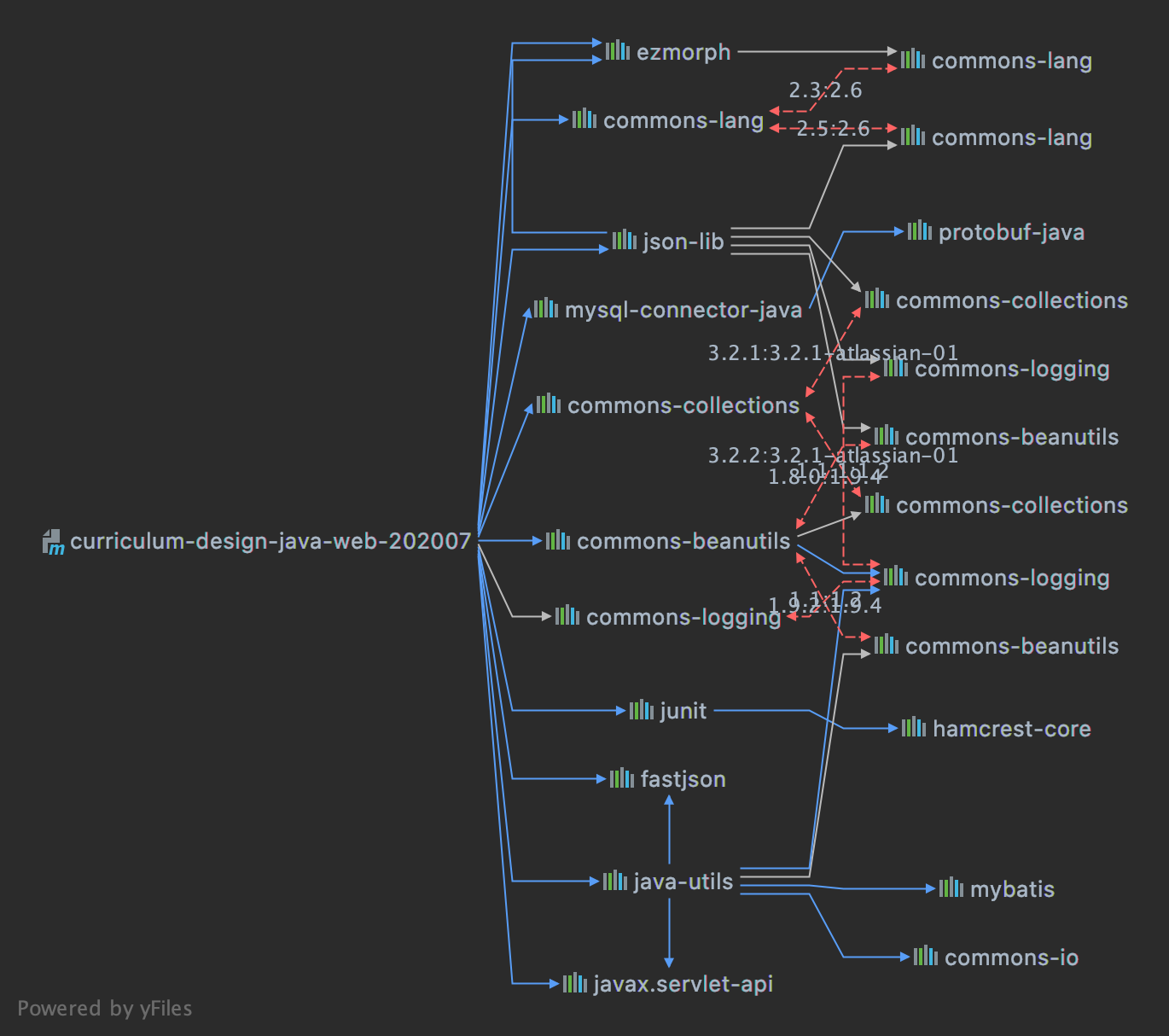


图 1第三方库依赖

### 分层分析

1. 驱动和中间介层
   1. 数据库驱动-JDBC

JDBC（Java Database Connectivity，Java 数据库连接）是一种用于执行 SQL 语句的 Java API 可以为多种关系数据库提供统一访问，它由一组用 Java 语言编写的类和接口组成。JDBC 提供了一种基准，据此可以构建更高级的工具和接口，使数据库开发人员能够编写数据库应用程序。

JDBC 是个“低级”接口，也就是说，它用于直接调用 SQL 命令。在这方面它的功能极佳，并比其它的数据库连接 API 易于使用，但它同时也被设计为一种基础接口，在它之上可以建立高级接口和工具。高级接口是“对用户友好的”接口，它使用的是一种更易理解和更为方便的 API，这种 API 在幕后被转换为诸如 JDBC这样的低级接口。

在关系数据库的“对象/关系”映射中，表中的每行对应于类的一个实例，而每列的值对应于该实例的一个属性。于是，程序员可直接对 Java 对象进行操作；存取数据所需的 SQL 调用将在“掩盖下”自动生成。此外还可提供更复杂的映射，例如将多个表中的行结合进一个 Java 类中。

随着人们对 JDBC 的兴趣日益増涨，越来越多的开发人员一直在使用基于 JDBC 的工具，以使程序的编写更加容易。程序员也一直在编写力图使最终用户对数据库的访问变得更为简单的应用程序。例如，应用程序可提供一个选择数据库任务的菜单。任务被选定后，应用程序将给出提示及空白供填写执行选定任务所需的信息。所需信息输入应用程序将自动调用所需的 SQL 命令。在这样一种程序的协助下，即使用户根本不懂 SQL 的语法，也可以执行数据库任务。

JavaSoft 提供三种 JDBC 产品组件，它们是 Java 开发工具包（JDK）的组成部份：JDBC 驱动程管理器、JDBC 驱动程序测试工具包和 JDBC-ODBC 桥。

JDBC 驱动程序管理器是 JDBC 体系结构的支柱。它实际上很小，也很简单；其主要作用是把 Java 应用程序连接到正确的 JDBC 驱动程序上，然后即退出。

JDBC 驱动程序測试工具包为使 JDBC 驱动程序运行您的程序提供一定的可信度。只有通过 JDBC 驱动程序测试的驱动程序才被认为是符合 JDBC 标准 TM 的。

JDBC-ODBC 桥使 ODBC 驱动程序可被用作 JDBC 驱动程序。它的实现为 JDBC 的快速发展供了一条途径，其长远目标提供一种访问某些不常见的 DBMS（如果对这些不常见的 DBMS 未实现 JDBC）的方法。

使用的时候，需要首先下载相应的驱动，然后配置 Java 工程里面的环境，再书写代码。因为 JDBC 是一个开放的标准，所以在使用的时候。需要先用反射加载驱动。可以使用 ClassLoader，也可以直接用 Class.forName 进行反射，反射的时候会自动运行驱动程序类的静态代码块，在静态代码块里面会进行数据库连接的初始化。同时因为这个关系，如果静态代码块初始出现异常，数据库功能将不能使用，除非重启程序。

1. 展示层
   1. Web 服务器- Tomcat

Tomcat 是 Apache 软件基金会（Apache Software Foundation）的 Jakarta 项目中的一个核心项目，由 Apache、Sun 和其他一些公司及个人共同开发而成。由于有了 Sun 的参与和支持，最新的 Servlet 和 JSP 规范总是能在 Tomcat 中得到体现，Tomcat 5 支持最新的 Servlet 2.4 和 JSP 2.0 规范。因为 Tomcat 技术先进、性能稳定，而且免费，因而深受 Java 爱好者的喜爱并得到了部分软件开发商的认可，成为目前比较流行的 Web 应用服务器。

Tomcat 服务器是一个免费的开放源代码的 Web 应用服务器，属于轻量级应用服务器，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试 JSP 程序的首选。对于一个初学者来说，可以这样认为，当在一台机器上配置好 Apache 服务器，可利用它响应 HTML（标准通用标记语言下的一个应用）页面的访问请求。实际上Tomcat 是 Apache 服务器的扩展，但运行时它是独立运行的，所以当你运行 Tomcat 时，它实际上作为一个与Apache 独立的进程单独运行的。

诀窍是，当配置正确时，Apache 为 HTML 页面服务，而 Tomcat 实际上运行 JSP 页面和 Servlet，另外 Tomcat 和 IIS 等 Web 服务器一样，具有处理 HTML 页面的功能，另外它还是一个 Servlet 和 JSP 容器，独立的Servlet 容器是 Tomcat 的默认模式。不过，Tomcat 处理静态 HTML 的能力不如 Apache 服务器。

* 1. JSP 的基础 Servlet 技术

Java Servlet 是 JSP 技术的基础，JSP 本身就是预先被编译成 Servlet，然后再运行的，而且大型的 Web 应用程序的开发需要 Java Servlet 和 JSP 配合才能完成，Servlet 这个名称大概源于 Applet，现在国内的翻译方式很多，本课程设计报告为了避免误会，本文直接采用 Servlet 这个名称而不做任何翻译，如果愿意，可以称之为“小服务程序”。Servlet 其实和传统的 CGI 程序和 ISAPI、NSAPI 等 Web 程序开发工具的作用是相同的，在使用 Java Servlet 以后，用户不必再使用效率低下的 CGI 方式，也不必使用只能在某个固定 Web 服务器平台行的 API 方式来动态生成 Web 页面。许多 Web 服务器都支持 Servlet，即使不直接支持 Servlet 的 Web 服务器也可以通过附加的应用服务器和模块来支持 Servlet。得益于 Java 的跨平台的特性，Servlet 也是平台无关的，实际上，只要符合 Java Servlet 规范，Servlet 是完全平台无关且是 Web 服务器无关的。由于 Java Servlet 内部是以线程方式提供服务，不必对于每个请求都启动一个进程，并且利用多线程机制可以同时为多个请求服务，因此 Java Servlet 效率非常高。但 Java Servlet 也不是没有缺点，和传统的 CGI、ISAPI、NSAPI 方式相同，Java Servlet 是利用输出 HTML 语句来实现动态网页的，如果用 Java Servlet 来开发整个网站，动态部分和静态页面的整合过程简直就是一场噩梦。这就是为什么 SUN 还要推出 Java Server Pages 的原因。

1. Model和View层
   1. Model和View层的关系

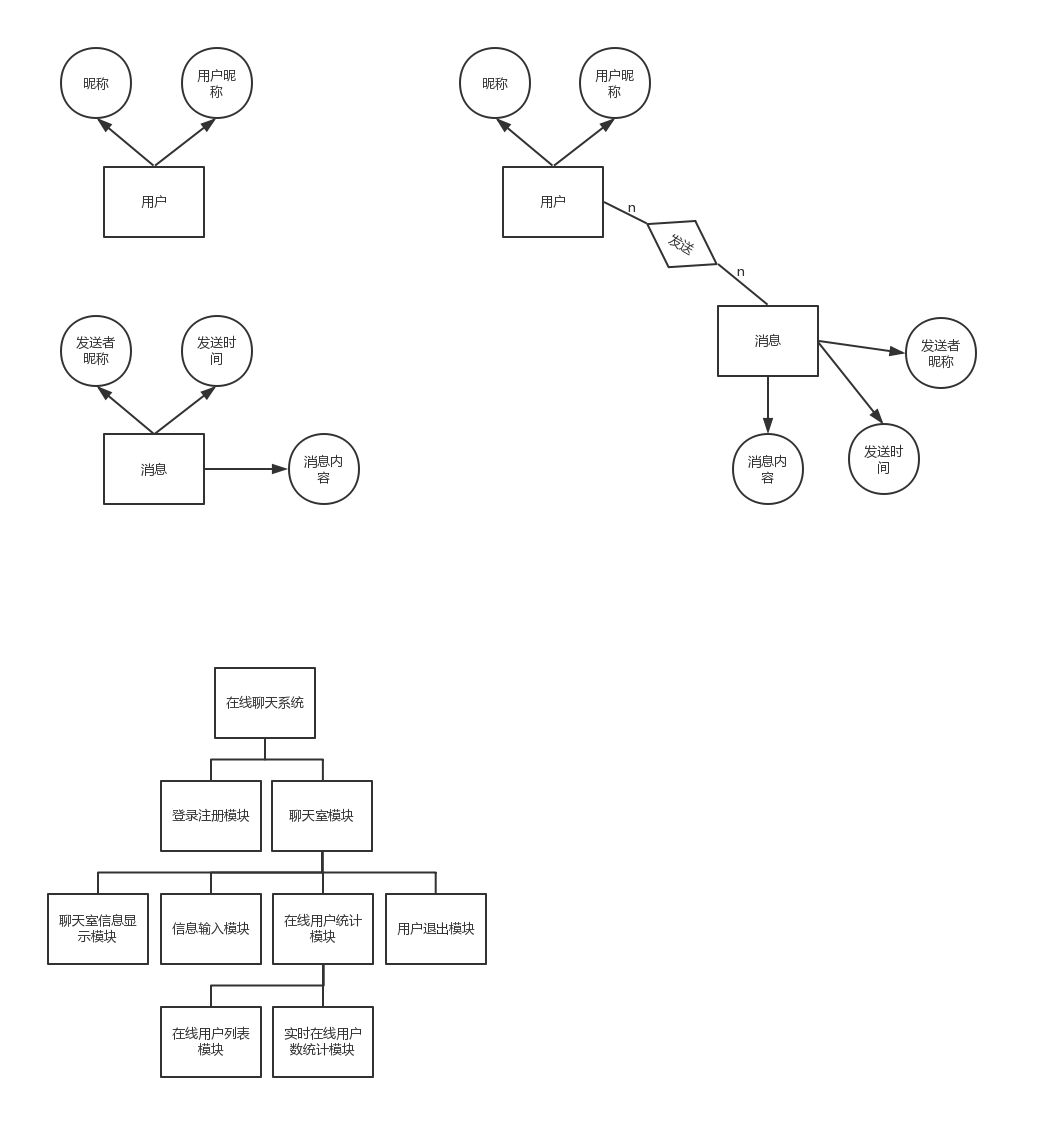


图 2 Model和View层的关系E-R图

1. Constant层

该层根据阿里巴巴Java代码设计在应用模块包下建立constant包，建立ConstConfig程序类存放应用级共享常量，这将可以消除编码过程中可能出现的绝大多数魔法值，使应用返回信息统一，实现一处修改多处同步，使程序复杂度降低。

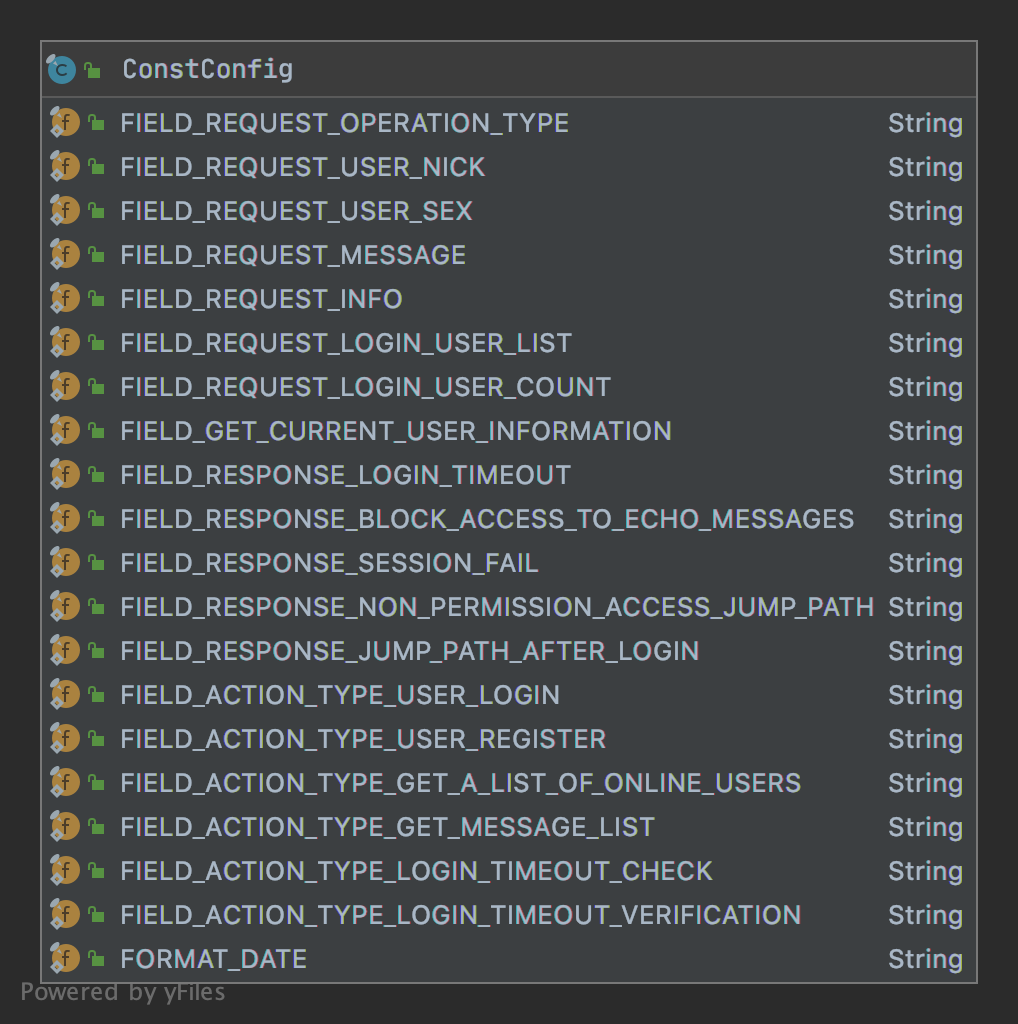


图 3 Constant层ConstConfig类图

1. DAO层

用户表：用来存储用户的基本信息，如用户昵称、性别等；

消息表：把每一条消息看做一个对象，用来存储消息的相关信息，如发送者 id、发送时间、消息内容等。

1. E-R图

以上表都是独立的实体，这些实体的 E-R 图设计为：

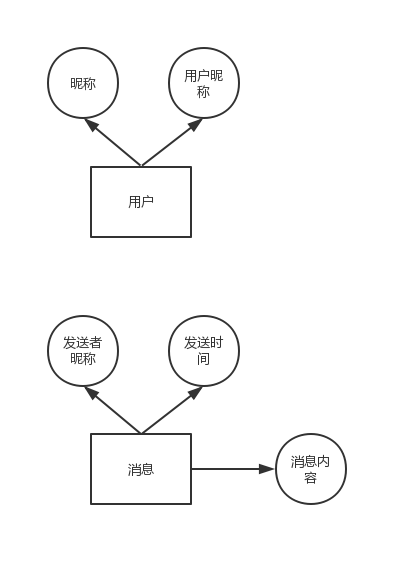


图 4实体E-R图

1. Service层

该层包含的主要是对 DAO 层的操作。这里的操作包含了对用户信息的查询。根据以上实体之间的关系，设计出总的 E-R 图。从图中我们可以清楚得看出每个实体之间的依赖关系。

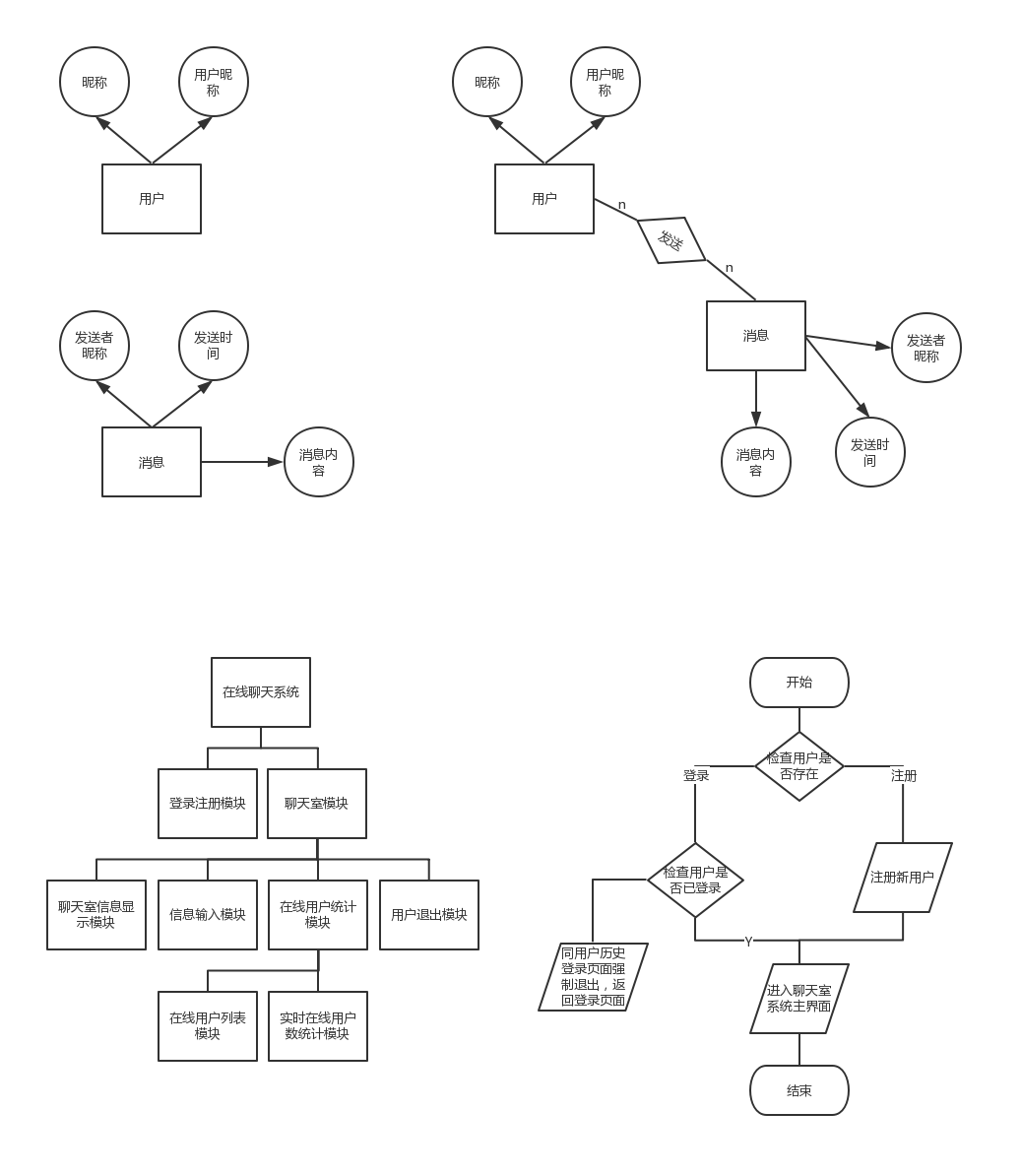


图 5 Service层E-R图

1. Controller 层
   1. 登录模块流程图

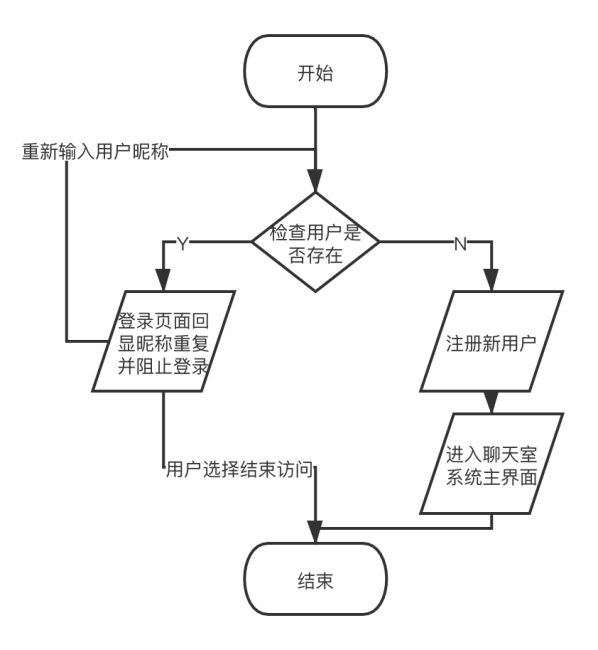


图 6登录模块流程图

1. UML

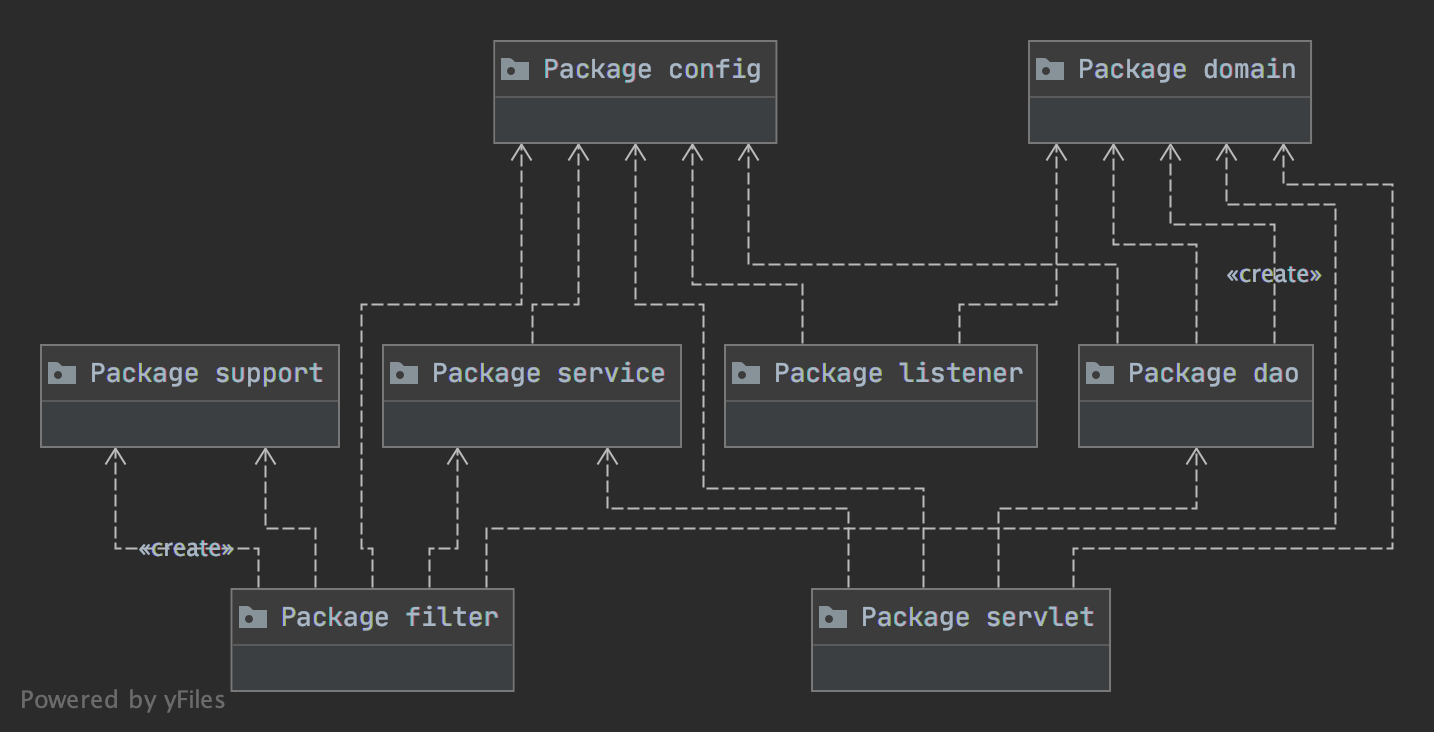


图 7程序包依赖图

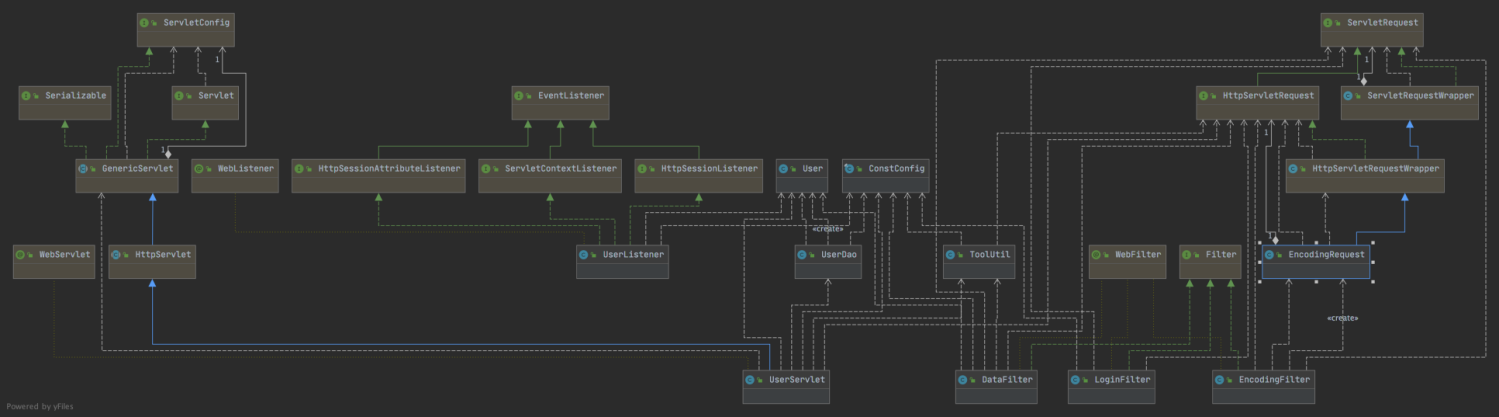


图 8 UML 类图

## 2、系统详细设计

### 前言

随着信息化技术在生活中的应用越来越广泛，网络即时聊天也逐渐成为人们的一种生活方式。本系统正是基于这样一种环境下应运而生。本系统为前台部分，主要实现用户的网上聊天业务流程、用户登录注册的功能。

本系统主要利用 Java Servlet 技术进行开发，有很强的逻辑性、可扩展性，便于维护。聊天室界面设计主要使用了 CSS进行聊天界面的美化。其次本系统设计还使用了 JSP 、JDBC 、JavaBean 、CSS、JavaScript、jQuery、AJAX等技术。

### 概要设计

1. 聊天室简要功能介绍

本次课程设计提供的简易聊天室是为许多喜爱聊天的用户开发设计的，和许多大型的网络聊天室相比该小聊天室功能比较简单，但能实现基本的聊天功能，点对多、点对点聊天和基本的用户功能，登录、注销、査看在线用户等。

1. 用户设置

功能描述：用户自定义自己的登录名，成功设置后，服务端能接收到新设置的用户名；

1. 登录功能

功能描述：在服务端开启的情况下，用户可以以自己自定义或者默认的登录名和登录地址端口登录到客户端，并能在服务器端显示登录状态；

1. 公聊

功能描述：用户在聊天室中可以选择与“所有人”聊天，聊天室中的其他在线用户可以正常接收并正确显示公聊消息；

1. 用户的状态

功能描述：每一个客户端用户都可以实时看到目前其他在线用户的状态；

1. 用户注销

功能描述：用户可以注销自己的账号，并重新登记登录名和地址；

1. 用户退出

功能描述：用户可以在正常情况下退出聊天室；

1. 文明用语控制

功能描述：当用户发送的消息被检测到不文明用语时，系统自动屏蔽相关关键词，客户端回显文本“\*\*\*”；

1. 服务端控制

功能描述：只有在服务端开启的情况下，用户才能登陆成功和聊天；用户访问的地址端口也必须与服务端的侦听端口一致，否则不能登录。

### 详细设计

1. 服务端设计
   1. 应用级常量
      1. ConstConfig.java
         1. 前台回传数据字段（input标签name属性名）名

|  |  |
| --- | --- |
| 常量名 | 常量值 |
| FIELD\_REQUEST\_OPERATION\_TYPE | "num" |
| FIELD\_REQUEST\_USER\_NICK | "nickName" |
| FIELD\_REQUEST\_USER\_SEX | "sex" |
| FIELD\_REQUEST\_MESSAGE | "msg" |
| FIELD\_REQUEST\_INFO | "info" |
| FIELD\_REQUEST\_LOGIN\_USER\_LIST | "userLoginList" |
| FIELD\_REQUEST\_LOGIN\_USER\_COUNT | "userCount" |

* + - 1. 获得Session中当前用户信息

|  |  |
| --- | --- |
| 常量名 | 常量值 |
| FIELD\_GET\_CURRENT\_USER\_INFORMATION | "user" |

* + - 1. 前台回显信息

|  |  |
| --- | --- |
| 常量名 | 常量值 |
| FIELD\_RESPONSE\_LOGIN\_TIMEOUT | "登录超时3分钟" |
| FIELD\_RESPONSE\_BLOCK\_ACCESS\_TO\_ECHO\_MESSAGES | "尚未登录，无权限访问" |
| FIELD\_RESPONSE\_SESSION\_FAIL | "sessionFail" |

* + - 1. 页面跳转路径

|  |  |
| --- | --- |
| 常量名 | 常量值 |
| FIELD\_RESPONSE\_NON\_PERMISSION\_ACCESS\_JUMP\_PATH | "../login.jsp" |
| FIELD\_RESPONSE\_JUMP\_PATH\_AFTER\_LOGIN | "../admin/index.jsp" |

* + - 1. 用户请求操作类型

|  |  |
| --- | --- |
| 常量名 | 常量值 |
| FIELD\_ACTION\_TYPE\_USER\_LOGIN | "0" |
| FIELD\_ACTION\_TYPE\_USER\_REGISTER | "1" |
| FIELD\_ACTION\_TYPE\_GET\_A\_LIST\_OF\_ONLINE\_USERS | "2" |
| FIELD\_ACTION\_TYPE\_GET\_MESSAGE\_LIST | "3" |
| FIELD\_ACTION\_TYPE\_SEND\_MESSAGE | "4" |
| FIELD\_ACTION\_TYPE\_LOGIN\_TIMEOUT\_VERIFICATION | "5" |

* + - 1. 日期时间格式

|  |  |
| --- | --- |
| 常量名 | 常量值 |
| FORMAT\_DATE | "yyyy-MM-dd HH:mm:ss" |

* 1. Tomcat web.xml 配置

本系统功能需求3指出的用户长时间未进行操作即自动退出系统，避免他人通过此用户的登录信息进行非法访问，造成其他用户使用本系统过程中使用体验产生不良影响。

本系统使用Session有效期超时销毁的方式完成此功能设计。具体编写代码需向web.xml 文件中添加如下代码：

<session-config>

<session-timeout>3</session-timeout>

</session-config>

* 1. Filter

本系统中Filter采用全新的Java EE 6 annotated class规范设计，使用@WebFilter注解标注于类声明上方，其中本系统需要使用到其中的两个属性——filterName和urlPatterns。这种方式可以给到开发者一种更加高效的代码开放方式，同时可以减少对web.xml的编写依赖。

* + 1. EncodingFilter.java
       1. 过滤URL路径：“/\*”
       2. 程序包路径：

com.kevinkda.univ.curriculum.design.b202007.java.web.chatroom.web.filter

* + - 1. 功能描述

此程序类依赖EncodingRequest.java程序类。

此程序类实现对Servlet获取前台提交信息的字符编码转换，由于Tomcat的默认字符编码是ISO-8859-1，而Java JDK环境的默认字符编码是UTF-8，在进行程序处理时会产生乱码，因此需要对字符编码进行转换。但是在所有的Servlet中都加入字符编码转换代码，则会增加程序代码的冗余，因此设计本类过滤所有的Request和Response达到全局字符编码转换目的。

* + 1. LoginFilter.java
       1. 过滤URL路径：“/admin/\*”
       2. 程序包路径：

com.kevinkda.univ.curriculum.design.b202007.java.web.chatroom.web.filter

* + - 1. 功能描述

此程序类实现在初始化时向ServletContext中添加于应用同生命周期的“userCount”用于存储全局的实时在线用户统计变量；在每次过滤链请求中对未登录用户的拦截操作。

某些用户在已知聊天室系统主界面URL的情况下，可能会未经登录直接访问，进而将可能造成系统异常、其他用户读取到脏数据，使其他用户使用体验造成不良影响，即本系统不允许使用匿名访问功能。

* + 1. DataFilter.java
       1. 过滤URL路径：“/User/\*”
       2. 程序包路径：

com.kevinkda.univ.curriculum.design.b202007.java.web.chatroom.web.filter

* + - 1. 功能描述

此程序类依赖WordAndTimeUtil.java程序类。

此程序类实现对用户提交发送的不文明用语进行拦截和屏蔽。

某些用户可能在聊天室发布不文明用户造成其他用户使用体验造成不良影响，同时违背国家相关法律的规定。因此本系统采用主动式文明用语检查规范设计，在用户提交数据到达本系统之时，主动过滤链系统启动，将用户不文明用语部分文字采用“\*\*\*”进行屏蔽，完成过滤后转交系统其他模块进行后续处理。

* 1. Listener

本系统中Listener采用全新的Java EE 6 annotated class规范设计，使用@WebListener注解标注于类声明上方。这种方式可以给到开发者一种更加高效的代码开放方式，同时可以减少对web.xml的编写依赖。

* + 1. UserListener.java
       1. 程序包路径：

com.kevinkda.univ.curriculum.design.b202007.java.web.chatroom.web.listener

* + - 1. 功能描述

此程序类实现对Session的创建和销毁的监听功能。Session创建时向控制台输出用户信息；Session销毁时发送用户长时间未操作页面跳转指令并核减实时在线用户数。

* 1. Servlet

本系统中Servlet采用全新的Java EE 6 annotated class规范设计，使用@WebFilter注解标注于类声明上方，其中本系统需要使用到其中的两个属性——filterName和urlPatterns。这种方式可以给到开发者一种更加高效的代码开放方式，同时可以减少对web.xml的编写依赖。

* + 1. UserServlet.java
       1. 访问URL路径：“/User/UserServlet”
       2. 程序包路径：

com.kevinkda.univ.curriculum.design.b202007.java.web.chatroom.web.servlet

* + - 1. 功能描述

此程序类是前台调用和提交数据的API接口。功能实现如下：

* + - * 1. 进行用户登录核查；

操作标识符：“0”

此操作将核验用户提交的昵称是否在系统中已存在，如已存在想用户返回核验标识符，如存在前台显示红色“昵称重复”标识；当不存在时前台继续提交AJAX请求进行用户注册请求。

* + - * 1. 进行用户注册；

操作标识符：“1”

此操作的前序操作为本API接口功能描述1，数据核验方式由前台JavaScript提供和控制[ 此核验方式存在被身份核验失效的风险，即攻击者可能通过修改AJAX传入参数的方式达到系统入侵目的，本系统因系统规模和容量暂不规避此风险。]。

该操作提供对没有注册过的用户提供注册，注册成功后向前台发送页面跳转指令至聊天系统首页。

* + - * 1. 获得在线用户列表；

操作标识符：“2”

此操作的前序操作为本API接口功能描述1，未登录用户将不被允许访问此操作。

该操作提供前台获取实时在线用户列表的方法。

* + - * 1. 获得消息列表；

操作标识符：“3”

此操作的前序操作为本API接口功能描述1，未登录用户将不被允许访问此操作。

该操作提供前台获取全局消息列表的方法，前台每隔1秒向本操作轮询一次，本操作将返回所有的。

* + - * 1. 发送用户提交的消息；

操作标识符：“4”

此操作的前序操作为本API接口功能描述1，未登录用户将不被允许访问此操作。

该操作提供将用户提交的消息保存入数据持久层中，并同步推送给当前在线的所有有效用户[ 即上一次操作与消息发送时间间隔不足3分钟，没有被系统强制退出的用户。]，后续用户也可以访问。

* + - * 1. 进行登录超时检查；

操作标识符：“5”

此操作的前序操作为本API接口功能描述1，未登录用户将不被允许访问此操作。

该操作提供前台长时间未访问的超时检查功能。超过时间未访问的用户将被自动注销登录。该操作返回超时检查核验结果。

* 1. Package Service
     1. WordAndTimeUtil.java
        1. 程序包路径：

com.kevinkda.univ.curriculum.design.b202007.java.web.chatroom.service

* + - 1. 功能描述

此程序类包含2项方法：日期格式处理和字符串屏蔽。

日期格式处理：根据传入的参数和日期格式进行格式化处理，本系统中传入的日期格式为"yyyy-MM-dd HH:mm:ss"，此功能的返回值将用于显示用户消息列表中用户消息发送时间的表示。

字符串屏蔽：根据传入的参数与数据库中不文明用语表进行精确匹配，如传入参数中包含不文明用语，其中不文明用语将会用“\*”代替，并覆写当前Request请求同名参数，转交后续程序继续处理。

* 1. Package domain
     1. User.java
        1. 程序包路径：

com.kevinkda.univ.curriculum.design.b202007.java.web.chatroom.domain

* + - 1. 功能描述

此程序类是用户实体类，用于处理对用户实体的操作。

此程序类包含两个字段：nickName（用户昵称）和sex（用户性别）。

此程序类包含重写的toString()方法和三个构造方法，以及对每个字段的Getter & Setter方法。

* 1. Package DAO
     1. UserDao.java
        1. 程序包路径：

com.kevinkda.univ.curriculum.design.b202007.java.web.chatroom.dao

* + - 1. 功能描述

此程序类2项方法：getIsLogin()和login()。

* + - * 1. getIsLogin()

此方法提供核验用户昵称是否进行了登录的服务，如用户传入的用户名已被登录则方法返回值为-1，阻止后续操作。

* + - * 1. login()

此方法提供经过核验后的用户昵称进行登录注册的操作，此方法将向数据库中添加传入的用户昵称，并向前台回传页面跳转指令进入聊天室系统主界面。

* 1. Package com.kevinkda.core.util.web.support
     1. EncodingRequest.java
        1. 程序包路径：

com.kevinkda.core.util.web.support

* + - 1. 功能描述

此程序类继承自HttpServletRequestWrapper类，通过覆写父类方法以将Filter中的原有ServletRequest对象进行替换，得到新的HttpServletRequest对象。通过此方法可以对Request对象进行更多功能的扩充。

此程序类在此处的方法是将用户从前台提交的ISO-8859-1字符编码转换为UTF-8字符编码。

1. 前台页面设计
   1. JavaScript

本系统中使用jQuery技术完成JS的快速开发。其版本使用jQuery 1.12.4，本系统开发过程中采用离线依赖，文件存储于web/js/目录下。

* 1. JSP
     1. 根目录
        1. index.jsp

此页面是项目首页，但此页面会自动跳转至login.jsp，进行用户登录。

* + - 1. login.jsp

此页面是用户登录注册页，提供给用户进行登录控制。

此页面提供的基于CSS样式的美化，给到用户良好的使用体验。页面背景采用CSS渐变色样式设计，具体代码如下：

background: linear-gradient(to top right, #5df5d5, #9ff8b8, #c8f3b1, #f4ffbb);

* + 1. admin目录
       1. index.jsp

此页面是聊天室首页，包含了对下述3个页面框架布局。

* + - 1. msg.jsp

此页面专用于框架布局中消息发送，此页面使用AJAX方式提交数据。

此页面提供的基于CSS样式的美化，给到用户良好的使用体验。页面背景采用CSS带透明值的纯色背景样式设计，具体代码如下：

background: rgba(126, 225, 255, 0.68);

* + - 1. msg\_info.jsp

此页面专用于框架布局中消息接受，页面使用AJAX技术每隔1秒向服务器进行轮询，以获得最新的消息列表。

此页面提供的基于CSS样式的美化，给到用户良好的使用体验。页面背景采用CSS渐变色样式设计，具体代码如下：

background: linear-gradient(to right, #91c9e8, #47a0e5);

* + - 1. user\_info.jsp

此页面专用于框架布局中在线用户用户列表和实时在线用户数的获取，页面使用AJAX技术每隔1秒向服务器进行轮询，以获得最新的在线用户列表列表。实时在线用户数会根据服务器回传数据实时核减。

此页面提供的基于CSS样式的美化，给到用户良好的使用体验。页面背景采用CSS纯色背景样式设计，具体代码如下：

background: #7ceefd;

# 四、系统的编码、测试和总结

## 1、系统代码

登录拦截功能代码：

@WebFilter(filterName = "LoginFilter", urlPatterns = "/admin/\*")

public class LoginFilter implements Filter {

@Override

public void destroy() {

}

@Override

public void doFilter(ServletRequest req, ServletResponse resp, FilterChain chain) throws ServletException, IOException {

HttpServletRequest request = (HttpServletRequest) req;

HttpServletResponse response = (HttpServletResponse) resp;

if (request.getSession().getAttribute(ConstConfig.FIELD\_GET\_CURRENT\_USER\_INFORMATION) == null) {

request.getSession().setAttribute(ConstConfig.FIELD\_REQUEST\_MESSAGE, ConstConfig.FIELD\_RESPONSE\_BLOCK\_ACCESS\_TO\_ECHO\_MESSAGES);

// 因为是目标地址是有admin的所以需要返回上一级别目录

response.sendRedirect(ConstConfig.FIELD\_RESPONSE\_NON\_PERMISSION\_ACCESS\_JUMP\_PATH);

} else {

chain.doFilter(req, resp);

}

}

@Override

public void init(FilterConfig config) throws ServletException {

// 初始化在线人数统计

int userCount = 0;

// 向全局存入

config.getServletContext().setAttribute(ConstConfig.FIELD\_REQUEST\_LOGIN\_USER\_COUNT, userCount);

}

}

字符编码转换过滤器代码：

@WebFilter(filterName = "SetEncodingFilter", urlPatterns = "/\*")

public class EncodingFilter implements Filter {

private static final String GET = "GET";

private static final String POST = "POST";

@Override

public void destroy() {

}

@Override

public void doFilter(ServletRequest req, ServletResponse resp, FilterChain chain) throws ServletException, IOException {

// 获得子接口的实例化对象

HttpServletRequest request = (HttpServletRequest) req;

// 设置响应字符编码集为UTF-8

resp.setCharacterEncoding("UTF-8");

resp.setContentType("text/html;charset=utf-8");

// 进行请求方法判断

if (GET.equals(request.getMethod())) {

// GET请求进行 request 对象的替换

EncodingRequest ecReq = new EncodingRequest(request);

// 传递过滤链，本次放行

chain.doFilter(ecReq, resp);

} else if (POST.equals(request.getMethod())) {

// Post请求，执行编码类型转换

req.setCharacterEncoding("UTF-8");

// 传递过滤链，本次放行

chain.doFilter(req, resp);

}

}

@Override

public void init(FilterConfig config) throws ServletException {

}

}

HttpServletRequest包装对象（EncodingRequest对象）设计代码：

/\*\*

\* 修改 {@link HttpServletRequest} 对象字符编码

\*

\* @author Kevin KDA on 2020/5/3 11:55

\* @version 1.0.0

\* @project TotalPro

\* @package com.kevinkda.core.util.web.support

\* @classname EncodingRequest

\* @apiNote <p></p>

\* @since 3.0.6

\*/

public class EncodingRequest extends HttpServletRequestWrapper {

/\*\*

\* 修改字符编码的 Request

\*/

private HttpServletRequest req;

/\*\*

\* 返回UTF-8编码结果

\*

\* @param name 传入需要获取的值的Key

\* @return java.lang.String 返回UTF-8编码结果

\* @author Kevin KDA on 2020/5/3 12:02

\* @description EncodingRequest / getParameter

\* @version 1.0.0

\* @apiNote <p></p>

\* @since 3.0.6

\*/

@Override

public String getParameter(String name) {

String value = this.req.getParameter(name);

try {

value = new String(value.getBytes(StandardCharsets.ISO\_8859\_1), StandardCharsets.UTF\_8);

} catch (Exception var4) {

System.out.println(var4.getMessage());

}

return value;

}

public EncodingRequest(HttpServletRequest request) {

super(request);

this.req = request;

}

}

## 2、系统的测试

软件测试的目的和原则

在开发软件系统的过程中，我们需要面对错综复杂的问题，不可避免的在每个生产阶段都会产生一些错误，我们力求进行严格的技术审查在每个阶段任务结束前尽可能多的发现和纠正软件中的错误。而软件测试的目的就是在软件投入生产环境前尽可能的修正软件中的错误，目前软件测试仍然是软件质量控制的最后防线。

软件测试分以下三种类型：

1. 单元测试：单元测试的用例从单元详细设计中导出。在单元测试中可以采用功能性测试和结构性测试两种；
2. 集成测试：在这个测试步骤中所发现的往往是概要设计的错误；
3. 系统测试：在这个测试步骤中所发现的往往是需求规格说明的错误。一般来说，系统测试是功能性测试，不是结构性测试。

本次设计进行了如下调试和测试，并提出相应的解决方案：

1. 访问失败

检查浏览器访问的路径是否为 http://localhost:8080/chatRoom/ ；

1. 前端脚本失效

检查是不是有语法错误导致后面的脚本全被屏蔽了；

1. 数据库连接出错

检查 JDBC 版本；检査数据库服务监听是否成功开启；检查数据库 URL、用户名密码等是否正确；

1. 配置文件加载失败

为 XML 的配置文件有些是 XSD 验证的，所以可能需要联网下载验证规则，此时应该保证网络畅通。其他就是可能配置文件书写的时候出现了语法错误；

1. Jdbctype 和 javatype 以及数据库 Type 问题

需要查询这三种 type 的对应关系，印象最深的就是不同数据库和 jdbctype 对应关系不一样。

# 五、总结

这次 Java Web课程设计是一门非常不错的课程，在此之前我写服务器基本都是自己造轮子自己写，但是这次课设使得我有机会使用现在还在使用JDBC + Servlet架构进行开发，同时，这次实验对我来说是一个巨大的挑战。因为之前只是对 JavaBean 的反射机制有所耳闻，原先用 Servlet的时候也是自己从GET和 POST处理函数为入口自己写处理，所以一开始对于 JavaBean 的反射还不是很熟练。但是，经过持续约一个月坚持不懈的努力，我终于完成了本次的实验任务。

当然在开发的过程中也遇到了很多问题。老师曾说，对于我们现在接触到的困难，之前肯定有人接触过，所以要好好利用互联网。课程设计，从安装 Tomcat，到配置数据库，到使用JDBC进行数据的增删改查，每步都曾遇到很棘手的问题，多亏了老师的帮助，也感谢在社区和论坛上帮我解决问题的朋友们。

采用B/S结构与JDBC框架。动态 WEB 中，程序依然使用客户端和服务端，客户端依然使用浏览器（Chrome、Safari、Firefox 等），通过网络（Network）连接到服务器上，使用 HTTP 协议发起请求（Request），现在的所有请求都先经过一个 WEB Server Plugin（服务器插件）来处理，此插件用于区分是请求的是静态资源（\*.htm 或者是 \*.html）还是动态资源。

当然，这次实验的另外一个体会就是，时间安排非常重要。老师在很早的时候就把任务书发了下来，但是当时我没在意，所以在之后做得非常紧。由此了解到，做好一个项目，每一阶段的时间都必须规划好。因为我们学生现阶段采用开发模式基本都是瀑布开发模式，就必须保证每一步的正确性和及时性。总之，本次设计锻炼了实践能力，増长了我发现问题解决问题的能力，大大丰富了自己的专业知识和技能。由于各种原因，系统的功能还没有得到进一步的完善，以后会继续努力，在今后的学习过程中，不断地提高自己。