**学号：2013040263**

**哈尔滨师范大学**

**学士学位论文**

**题 目 基于java的仿QQ和QQ空间的设计与实现**

**学 生 李林峰**

**指导教师 石晔琼**

**年 级 2013级**

**专 业 软件工程**

**系 别 软件工程**

**学 院 计算机科学与信息工程学院**

哈尔滨师范大学

2017年5月

**学 士 学 位 论 文**

**题 目 基于java的仿QQ和QQ空间的设计与实现**

**学 生 李林峰**

**指导教师 石晔琼**

**年 级 2013级**

**专 业 软件工程**

**系 别 软件工程**

**学 院 计算机科学与信息工程学院**

**摘要：**本课题主要研究基于java的仿QQ聊天软件以及QQ空间系统，模拟实现腾讯QQ的基本的聊天软件功能，诸如注册、登录、添加好友、好友聊天、文件传输等一些列功能，其次模拟实现QQ空间的常用功能，诸如发表说说、上传相册、写日志，留言等功能，以更好的用户体验性为目的开发出界面简洁，功能适用的即时通讯聊天互动软件。

本文首先描述了聊天软件的背景和发展趋势，其次是课题的目的及意义等分析，然后介绍了软件开发的环境和开发工具及其技术，给出了软件的系统架构，各个模块的流程图及用例，然后详细的介绍了软件的设计与实现，最后对系统进行了简单的测试。

**关键词：**java；QQ聊天软件；QQ空间；用户体验

**目录**

[第一章 绪论 1](#_Toc1599)

[1.1 社交软件的发展背景和趋势 1](#_Toc7824)

[1.2 课题研究目的及意义 1](#_Toc19680)

[1.3 可行性分析 2](#_Toc4249)

[1.4 研究内容和方法 2](#_Toc31778)

[第二章 系统开发工具及相关理论概述 3](#_Toc27991)

[2.1 UML技术 3](#_Toc13944)

[2.2 RUP技术 3](#_Toc7239)

[2.2.1 RUP概述 3](#_Toc8436)

[2.2.2 RUP概述的三个核心 4](#_Toc17301)

[2.3 Eclipse 5](#_Toc4330)

[2.4 MySQL 5](#_Toc29618)

[2.5 Tomcat 5](#_Toc27568)

[2.6 Spring 5](#_Toc24738)

[2.7 Mybatis 6](#_Toc28984)

[2.8 Socket 6](#_Toc7842)

[第三章 系统需求分析 9](#_Toc12438)

[3.1 功能性需求 9](#_Toc17683)

[3.2 非功能性需求 12](#_Toc7610)

[3.2.1 安全型需求 12](#_Toc12765)

[3.2.2 易用性和易维护性需求 12](#_Toc14812)

[3.2.3 性能需求 12](#_Toc14487)

[第四章 系统的设计与实现 13](#_Toc3184)

[4.1 系统总体设计 13](#_Toc9270)

[4.1.1 逻辑结构设计 13](#_Toc22689)

[4.1.2 开发环境 14](#_Toc26399)

[4.2 基于UML的系统模型构建 14](#_Toc7063)

[4.2.1 系统主要模型设计 16](#_Toc29894)

[4.3 数据库表设计 22](#_Toc546)

[4.3.1 表结构和表间关系设计 23](#_Toc22850)

[4.3.2 数据连接池 27](#_Toc14458)

[4.3.3 数据缓存 28](#_Toc1658)

[4.4 系统主要功能模块的设计与实现 28](#_Toc19760)

[4.4.1 QQ用户注册与登录 29](#_Toc19423)

[4.4.2 QQ好友添加与删除 31](#_Toc26763)

[4.4.3 QQ好友上下线提示 32](#_Toc21710)

[4.4.4 QQ好友多对多聊天 32](#_Toc25570)

[4.4.5 QQ个人信息管理 33](#_Toc11012)

[4.4.6 QQ空间说说评论管理 33](#_Toc26802)

[4.4.7 QQ空间相册管理 34](#_Toc26071)

[4.4.8 QQ空间留言管理 36](#_Toc15751)

[第五章 系统测试 37](#_Toc13326)

[5.1 系统测试用例设计 37](#_Toc22245)

[5.1.1 单元测试 37](#_Toc7559)

[5.1.2 集成测试 42](#_Toc13534)

[第六章 总结和展望 43](#_Toc786)

[致谢 44](#_Toc6649)

[参考文献： 45](#_Toc22061)

**[Abstract](#_Toc20106)** [46](#_Toc20106)

# 第一章 绪论

众所周知，在社交软件风靡全球之前，人们传统的联系方式无外乎电话和短信，以及曾经兴起一时但很快被淘汰的飞信。随着互联网技术兴起后，国内外各类社交互动类软件如有后春笋般相继推出，即时便捷人性化的通讯方式大大的丰富了人们的日常生活。

## 1.1 社交软件的发展背景和趋势

国内的QQ兴起后，YY、微信、陌陌等同类软件相继进入市场，为了尽可能便利于用户，从PC端到移动端，各种人性化功能层出 不穷，各种社交软件的用户量之大，认可度之高，远胜于其它互联网软件产品。

伴随着互联网技术的发展，QQ率先成为国内崛起的一款即时通讯软件，多年来一直与时俱进的跟随着人们需求而改进着，因此也已成为人们日常生活中不可或缺的交流工具。虽然近几年有各式各样的社交软件出现，但也并没有破灭其老大哥的地位，因为新功能的上线总是能给用户带来一波又一波的惊喜。  
 微信是QQ崛起之后由腾讯开发的另一款比较高端大气的熟人社交软件，通过漂流瓶、摇一摇、附近的人等方式与陌生人交友，这些方式几乎都能够抓住人们具有童心而又惊喜感的交友方式，让人们玩的不亦乐乎，更绝配的是还可以录制一些小视频，分享到朋友圈。  
 微博一种用于用户信息共享和传播、以及获取的社交平台应用，也可以理解为就是博客的微型信息传播，用户以140字（包括标点符号）的文字+图片分享精彩的信息。  
 GaGa是基于翻译的国际社交软件，提供全球主流的八种语言九种文字的实时在线人工智能翻译服务，让用户之前的交流能够通过即时翻译沟通无限。  
 派派是一款以语音为主的社交软件，用户之间都是通过语音进行交流的，给人一种真实的感觉。同时通过建立30人的家族，进行各种PK争霸，完全可以做到人人都上麦的语音聊天。  
 陌陌是一款基于地理位置的陌生人交友软件，实现了人们线上聊天交友，线下邀约的真实交友模式，完全的实现了真实交友的目的。

Facebook,Myspace,twitter，Skpyle，GaGa这是四大国外社交软件深受外国人的喜爱，不过Facebook是最受外国人喜欢的一款社交软件，不过也已经可以实现多种语言的翻译了。然而后来者GaGa以绝对的翻译为基础的社交模式崛起，目前很受全世界人民的喜爱。所以国外在语言问题上似乎更进了一步，这也是国内社交软件也要攻克的问题。

随着越来越多的社交软件比肩而行，只有功能的个性化能始终抓住绝大多数用户感性倾斜的方向，不断去旧迎新，不断升级改进，才能始终跟上社交类软件发展的潮流，总得来说，QQ是比较成功的。

## 1.2 课题研究目的及意义

当下流行的QQ软件的开发采用的是C++语言，javaSE桌面开发过去曾被大规模使用过，但由于其开发效率问题逐渐被C#和C++所替代，但并不等同于说javaSE不能很好的开发桌面应用程序，此文将采用javaSE作为仿QQ聊天软件开发的工具进行开发，结合javaEE技术开发仿QQ空间系统。

国内QQ聊天软件的诞生，给我们的生活带来了极大便利，于是社交软件缘何兴起与如何服务于用户就变得很有意义了，里面涉及到的互联网技术尤为值得探讨，网络编程运用广泛，社交软件是一个很好的运用实例，一个好的实例可以使得技术能延伸到更多的领域。

QQ空间，类如博客系统一样，属于网络上自己的资源信息领地，同时可以和好友进行信息资源共享互动。QQ空间系统伸缩性很好，简单的说，可以只有说说、相册、日志、留言等基础功能，一旦绑定到QQ用户，能服务于用户的功能就可以无限延伸设置了，常见的如空间里的各种用户互动，QQ空间游戏等集中嵌入其中，这使得原本相对静止单一的博客系统变得更加生动有趣，这对用户的吸引力就增添了不少砝码，在互联网软件和系统日益风靡的现在，一款产品设计的好坏能决定产品的收益。所以，对QQ空间的研究和实践是很有实际意义的，尤其是和QQ软件的绑定。

现如今，各种系统集成，云系统，云平台孕育而生，系统与系统间互相访问，资源信息的共享对用户的体验感来说尤为重要，各种传统封闭式的独立软件系统也会越来越少。综上所述，探讨并模拟开发一款QQ和QQ空间系统是很有意义的。

## 1.3 可行性分析

Java作为一门经典的面向对象的编程语言，吸取了C++的众多优点，另外其跨平台的特性更是使得它风靡全球，java体系庞大，分支众多，桌面程序、web应用、手机移动端、嵌入式程序等随处可见它的影子。

本设计仿QQ桌面程序可以采用javaSE的swing包开发界面，通信采用基于java的Socket编程，QQ空间系统采用javaEE体系相关技术进行开发。

综上所述，本设计是可行的。

## 1.4 研究内容和方法

本系统分为两大模块，QQ聊天软件部分为桌面版程序，采用C/S架构；QQ空间系统为WEB应用程序，采用B/S架构，基于MVC模式开发。

QQ聊天软件部分：采用JavaSE开发，界面采用Swing包开发，信息交互采用基于tcp/ip协议的Socket编程，文件流处理采用I/O流，服务端提供数据持久服务供客户端调用，整体采用spring结合mybatis框架，涉及到客户端编程的采用HttpClient补充。

QQ空间系统部分：采用JavaEE开发，框架搭建采用Spring容器框架集成控制层WEB框架SpringMVC和持久层框架Mybatis开发，服务端辅以Servlet、Filter、Jsp为主、持久层缓存采用Mybatis缓存机制，数据库采用Mysql，服务器采用Tomcat。界面前端采用Html5、响应式CSS框架Bookstrap、JavaScript库采用Jquery，交互端技术采用基于Jquery的AJAX，数据格式Json等。

# 第二章 系统开发工具及相关理论概述

根据系统开发的目的和要求，本项目使用面向对象软件工程方法学指导软件系统的设计与实现。使用统一建立模型工具UML和统一过程RUP融合的方法,运用eclipse开发工具以及mysql数据库等完成。下面对主要相关理论与技术进行阐述说明。

## 2.1 UML技术

UML是一种统一建立模型或[规范建立模型的工具](http://baike.baidu.com/item/%E6%A0%87%E5%87%86%E5%BB%BA%E6%A8%A1%E8%AF%AD%E8%A8%80" \t "http://baike.baidu.com/_blank)，它可以说是一个支持模型化和软件系统开发的图形化语言，为软件开发的所有阶段提供模型化和可视化支持，包括由需求分析到规格，到构造和配置[5]。

UML规范拿来阐述建立模型的概念有很多，比如类，在这是基于对象的、关联、职责、行为、接口、用例、包、顺序、协作，以及状态[6]。

规范建立模型的工具UML的主要内容能够由五类图来表示，分别是用例图、静态图、行为图、交互图和实现图。

## 2.2 RUP技术

### 2.2.1 RUP概述

RUP是统一软件开发过程的一种方法论。它以网络编程的程序开发为基础，体现的是一种面向对象的思想。和它相似的其他产品有OOSP，这也是一种面向对象的思想，还有OPEN Process,它们都把开发过程中的面向过程的思想和开发的组件融合在一个框架内。还有很多其他的软件开发过程，与它们比起来，RUP保存了自己独有的特点，它是通过用例来驱动、反复、递增的。

它的核心的工作流一共有9个见下图：

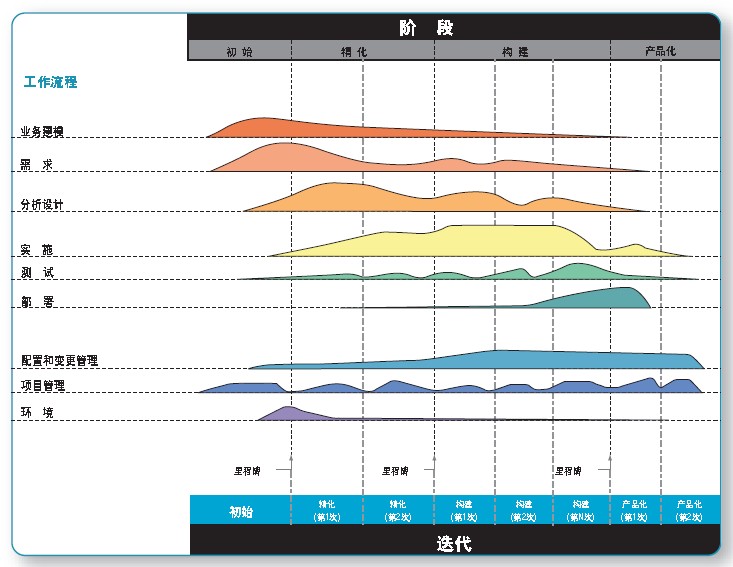
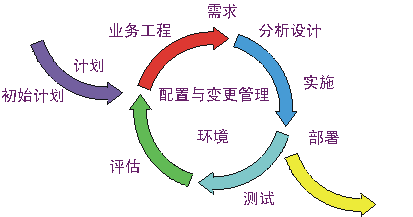


图2.1 RUP开发过程

模型是产品组成的各个过程成分，也是下一个过程成分的工作基础。RUP过程中，我们可以使用UML工具来对用例模型、设计模型（对象模型）等模型进行表达，不过，这不是绝对的，也有各种各样的方式可以表达这些模型。RUP继承了原型法的核心，采用螺旋上升的模式进行开发。在所有软件系统的开发中，一般都是反复，渐增，循环的，就如下图一样：

图2.2 RUP迭代式开发过程

### 2.2.2 RUP概述的三个核心

1、用例驱动（用例和风险驱动）

建立模型是RUP的一个重要组成部分，建立模型的一个方法：用例建模是它推崇的一个比较有效的建立模型的方法。全部的用例组合在一块组成了所要的用例模型。它摒弃了传统观念上的系统方面的功能，主要体现的是系统的所有的功能，并且最重要的一点是它可以应用为客户与开发者间的一份有效合同。

RUP的用例驱动体现的是从定义到设计再到测试的一个用例过程。这体现出一个开发过程的开始阶段是用例定义出的一个系统，用例最初先被确立，之后被设计，最后一步就又变为了测试。

用例不仅能有效的表达功能性需求，还驱动软件的设计、实现和测试，使软件产品充分地满足用户的需求。

2、体系架构

根据RUP的描述，我们将使用这些不同的图，从5个不同的角度来建立模型。软件建立的过程中，不同编码人所注重的方面是不一样的，因而软件体系是多维的。RUP应用4+1的模型分别是逻辑视图、开发视图、处理视图、物理视图和场景视图，如下图所示：

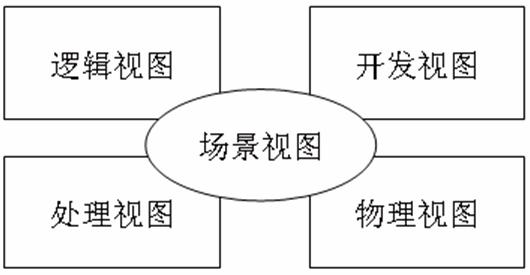
****

图2.3 RUP 4+1视图模型

3、迭代递增式开发

软件开发采用反复和递增主要有以下几大特点：

（1）处于软件开发的初期时候就可以对重点的，影响比较大的潜在问题进行处理；

（2）可以提出一个软件体系结构来指导开发；

（3）可以比较好的解决不可规避的需求变更；

（4）可以较早的得到一个可运行的系统，鼓舞开发团队的士气，增强项目成功的信心；

（5）为开发人员提供一个更有效工作的开发过程。

## 2.3 Eclipse

支持JAVA的开发工具比较多，目前，市面上最流行的IDE依旧是Eclipse，它具有与其他IDE共有的如项目结构快速构建、项目调试、开发提示等基本功能，相对于自带集成了大量插件Myeclipse来说，它要小巧很多，自带插件比较少，对于开发者而言，不管是做J2SE开发还是J2EE开发，甚至是做安卓SDK开发，安装配置插件也是开发人员的基本功，由于Eclipse本身比较小巧，所以开发效率上也比较高，在开发的过程中，通常是需要安装配置何种新插件，开发人员才去进行安装配置。

## 2.4 MySQL

Mysql是一款小型数据库，小巧易用，安装配置简单。相对于Orcale、DB2等大型数据库的复杂和昂贵，在成本控制上它更值得选择，相对于Sql Server系统平台局限的开放性（windows的产物），它又显得更容易扩展。虽然在安全和性能上相比Oracle显得逊色，但是在开发中小型软件的时候，综合考量Mysql无疑是不错的选择。

## 2.5 Tomcat

Tomcat是一款非常流行的免费开源的Web服务器，来自Apache基金组织，在Java重量级框架EJB开发的时代，JBoss服务器是EJb的标准容器，而Jsp和Servlet技术兴起后，Tomcat、Weblogic等Web服务器相继流行，作为Jsp和Servlet的容器，在目前流行的Java轻量级框架中Tomcat更是充当Web服务器的主要角色，对Spring体系有非常好的兼容性。

## 2.6 Spring

Spring框架可以作为是java轻量级框架的代表，大大简化了过去重量级EJB才能完成的事情，spring的核心有两个，即IOC（控制反转）和AOP（面向切面编程），spring的IOC通过java反射机制和注解结合XML解析，对配置的bean对象进行装配，来完成javabean对象的创建和bean关系的维护；spring的AOP是一种高效于面向过程和面向对象的思想，通过预编译和动态代理的方式可以对一批对象进行编程，应用已十分广泛。Spring容器大大降低了程序的耦合度，提高的程序的重用性，同时也提高了程序的开发效率。

Spring体系尤为庞大，子项目和子应用有springMVC、spring web flow、spring security、spring task、spring Cache等等，其中springMVC是非常优秀且运用广泛的web框架，相对于纯servlet和struts2框架，在线程安全和效率，以及性能方面更胜一筹，servlet对多线程并发的控制尤为复杂，而struts2本身是基于类的，每个Action请求都会实例化一个Action对象，本身是非单例的，所以线程是安全的，但同时也预示着效率的降低，而springMVC不一样，Controller类里面都是基于方法的操作，虽然是singleton的，但一般情况下也能保证线程的安全。

## 2.7 Mybatis

如果我们将ORM框架Hibernate比作全自动的持久化框架的话，那么无疑Mybatis（前身ibatis）就是半自动化持久层框架，hibernate提供了自己的查询语言HQL（hibernate Query Language），直接屏蔽了不同数据库种类对应不同sql语句这一层，对于数据库类型的切换尤为方便，使开发者可以不关注具体的sql如何编写。而Mybaits虽然使用的是纯sql语句进行持久化操作，也不能像hibernate那样自由的切换数据库，但是同hibernate一样，也是对jdbc做了轻量级的封装，但是配置和使用却更加简捷。应付大多数程序应用也是没有什么问题的，由于其学习和使用的简捷和高效性，如今被广泛应用。

Mybatis框架的底层实现也尤为简单，ORM框架基本都会通过一个映射文件来关联java实体类对象和对应的数据库表记录，框架采用xml解析配置文件，对相应的sql语句进行jdbc重构操作，以达到数据持久的目的。

## 2.8 Socket

Socket，英文名是插座的意思，Socket编程即网络编程，所谓网络编程，是指两个应用程序通过一个双向的通信管道实现数据交互。

人和人通话的前提是你得指定和你通话的人是谁，以及你们得说一种语言（协议），这样才能顺利完成交流。计算机网络也是如此，首先你需要指定你和哪一台机子的哪个应用程序通信，然后指定通讯的协议，这样才能完成接下来的数据交互。所以我们需要简单介绍以下几个概念:

Ip：ip是Internet Protocol的外语缩写，即网络间的互联协议。Ip地址由4个字节来表示，每个字节代表一段，每一段最大数255，这就是IPV4，Ip地址唯一标识一台计算机。

端口：可以这么理解，一个占用的端口指向运行在计算机上的一个应用程序。

TCP/UDP协议：两种网络通信协议，建立在IP的基础上。TCP协议的特点是，首先得确认和对方建立了连接，之后每一次数据传输都需要保证对方给于回应，如果对方不回应，则阻塞在这里一直等待对方回应。所以，TCP协议是可靠的，但是也是低效的；相反UDP协议是不可靠但高效的，它提供了一种发送封装数据的原始IP数据报的方法，且传输数据无需建立连接，只管将数据传输给对方，但对方能否接收到是未知的。二者各有千秋和不足，要求安全可靠但不保证高效率的通信采用TCP协议，要求高效但不保证安全可靠的通信采用UDP协议。

于是，网络通信首先通过IP地址加端口号指定了将要与之进行通信的应用程序，然后通过TCP或UDP协议进行通信。

Java给我们提供了Socket编程的API，我们采用的是TCP/IP协议，ServerSocket类指定服务端监听端口，然后调用一个阻塞式的方法accpet()进行连接等待，等待客户端与之建立连接，同时返回一个Socket类型的对象，这个socket对象可以形象的理解为一根通信管道；客户端创建一个socket对象，指定要与之通信的应用程序(服务端的ip+端口)，这样就和处在等待连接状态的服务端建立起了连接，此时，服务端返回的Socket对象和客户端创建的Socket对象便组成了一根双向管道，以便服务端和客户端进行数据传输。如下图所示：

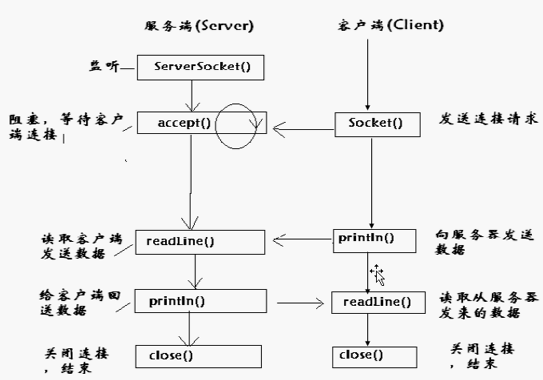


图2.4 基于TCP/IP协议的Socket通信示意图

数据传输的过程中，网络应用程序的数据到网络最底层有一个分层的思想，这不是此文论述的重点，但是可以做一个了解，数据传输的时候，需要将数据进行一个封装，由应用层逐层到底转化为0101010的数据传递到目标机器特定的应用程序，目标应用程序接受到数据后，进行一个数据拆封，从0101010数据逐层向上转化为应用层数据抵达应用层。如下图所示：

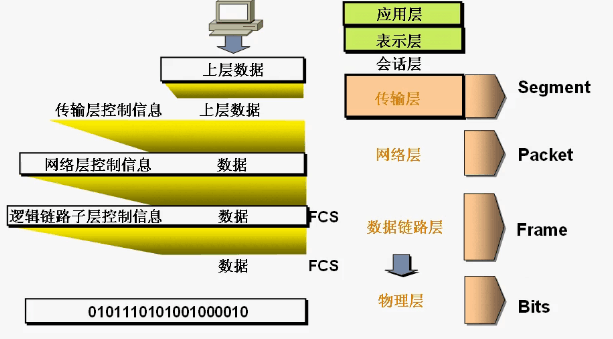


图2.5 Socket通信数据封装示意图

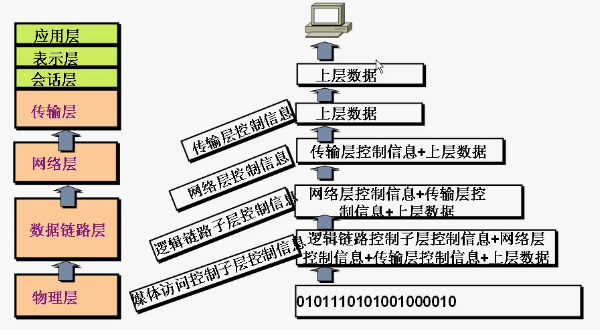


图2.6 Socket通信数据拆封示意图

# 第三章 系统需求分析

在互联网技术日益发展的今天，产品设计基于一个完整的需求分析，需求分析是软件开发过程中非常重要的一步，改善企业内部各个供应链各个环节的管理，做到资源适度调配，及时适应用户变更的新需求，才能获得更多的市场机遇。

本设计使用javaSE技术编写一个桌面版仿QQ聊天软件，实现用户注册、登陆、添加删除好友、与好友聊天、消息记录管理等功能，使用javaEE相关技术编写一个仿QQ空间系统，实现发表说说与评论、管理空间说说动态、上传照片与相册管理、发表日志与日志管理、发表留言与留言管理、访问权限设置等功能。系统结构如下图所示：

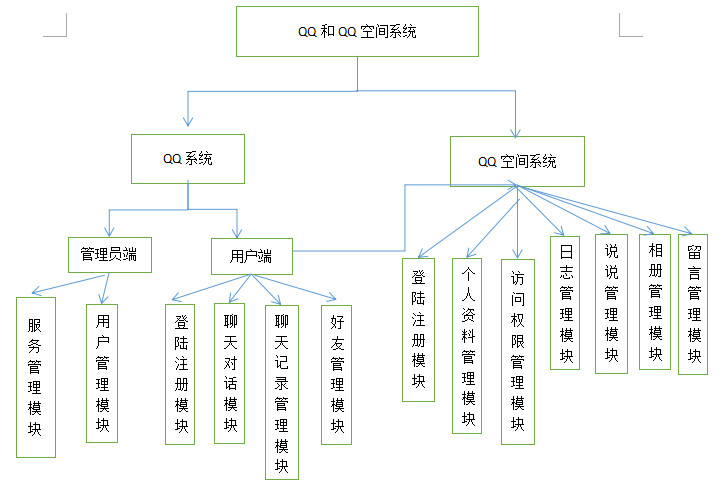


图3.1 系统结构图

## 3.1 功能性需求

本设计分为两个子项目，仿QQ聊天软件基于C/S构架，仿QQ空间系统采用B/S构架，功能从用户、角色、权限的角度具体说明如下：

用户：QQ服务端管理员、QQ客户端用户

角色：管理员、用户、游客

权限：依据角色定义权限如下：

管理员：启动或关闭QQ服务器、查看QQ用户及信息、强制用户下线。

普通用户：QQ号注册、登陆、个人信息管理、添加或删除好友、查看好友信息、与好友聊天(支持多对多聊天)、聊天记录管理、访问自己QQ空间、访问好友QQ空间（有访问权限的好友）、发表说说、发表说说评论、发表留言、删除说说、删除说说评论、删除留言等功能、其中，QQ空间包括访问权限管理、日志管理、说说管理、相册管理、留言板管理、评论管理、个人档信息管理等，具体功能需求参考表3.1功能需求表:

表3.1 系统功能需求

|  |  |  |
| --- | --- | --- |
| **功能名称** | **功能说明** | **操作角色** |
| 服务管理 | 启动服务 | 管理员 |
| 关闭服务 | 管理员 |
| 用户管理 | 分页查看QQ用户列表 | 管理员 |
| 查看QQ用户信息 | 管理员 |
| 强制用户下线 | 管理员 |
| 注册登录 | 注册QQ账号 | 游客 |
| 登录QQ | 用户 |
| 登录QQ空间 | 用户 |
| 个人信息管理 | 查看QQ个人信息 | 用户 |
| 编辑QQ个人信息 | 用户 |
| 查看QQ空间个人信息 | 用户 |
| 编辑QQ空间个人信息 | 用户 |
| 好友管理 | 通过QQ号查找QQ用户 | 用户 |
| 通过QQ昵称关键字查找QQ用户 | 用户 |
| 添加QQ好友（查找结果） | 用户 |
| 删除QQ好友 | 用户 |
| 查看好友QQ个人信息 | 用户 |
| 访问好友QQ空间（有访问权限的） | 用户 |
| 上下线提示 | 好友上线提示 | 用户 |
| 好友下线提示 | 用户 |
| 聊天管理 | 选择好友与之聊天 | 用户 |
| 查看与好友的聊天记录 | 用户 |
| 权限设置 | 设置QQ空间访问权限（设置哪些好友可访问） | 用户 |
| 日志管理 | 发表日志 | 用户 |
| 分页查看日志列表 | 用户 |
| 阅览日志 | 用户 |
| 删除日志 | 用户 |

续表3.1 系统功能需求

|  |  |  |
| --- | --- | --- |
| **功能名称** | **功能说明** | **操作角色** |
| 说说管理 | 发表说说 | 用户 |
| 查看说说列表 | 用户 |
| 删除说说 | 用户 |
| 评论管理 | 发表说说评论 | 用户 |
| 查看说说评论 | 用户 |
| 删除说说评论 | 用户 |
| 相册管理 | 新建相册 | 用户 |
| 查看相册列表 | 用户 |
| 编辑相册 | 用户 |
| 删除相册 | 用户 |
| 上传照片到指定相册（批量上传） | 用户 |
| 分页查看照片列表 | 用户 |
| 浏览照片大图（上一张/下一张） | 用户 |
| 删除照片 | 用户 |
| 留言管理 | 发表留言 | 用户 |
| 分页查看留言列表 | 用户 |
| 删除留言 | 用户 |
| 退出登录 | QQ退出登录 | 用户 |
| QQ空间退出登录 | 用户 |
| 用户管理 | 冻结QQ用户账号 | 管理员 |

针对上述功能需求图表做以下特别说明：

1：访问自己QQ空间的时，可以编辑个人档信息、发表说说、发表说说评论（支持评论自己所发表的说说）、删除说说、删除说说评论、发表日志、删除日志、发表留言（支持给自己留言）、删除留言、新建相册、编辑相册、删除相册、上传照片、删除照片。

2：访问好友QQ空间（有访问权限的好友QQ空间）时，可以查看好友的日志、说说、评论、相册、照片、留言、个人档信息、也可以发表说说评论。

3：QQ管理员冻结QQ用户账号后，QQ用户将无法再登陆QQ使用，也无法再访问好友的QQ空间，但是该QQ用户仍然可以登录自己的QQ空间。

4：评论功能仅仅针对说说提供，相册、日志和留言不支持评论功能。

本设计将仿QQ聊天软件子项目和仿QQ空间系统子项目分为三个大的模块，分别为QQ管理员功能模块、QQ用户功能模块、QQ空间系统功能模块。

QQ管理员功能模块和QQ用户功能模块同属于仿QQ聊天软件，属C/S架构，QQ用户功能模块和QQ空间系统功能模块相交互。QQ空间系统可以直接通过QQ用户功能模块进入。

QQ管理员功能模块：主要分为服务管理和用户管理两大模块，前提是管理员登陆到系统。如下图所示：

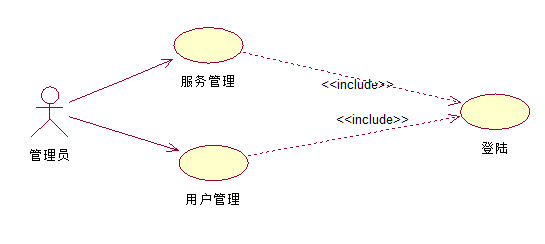


图3.2 QQ聊天软件管理员功能用例图

QQ普通用户功能模块：主要分为用户注册登陆、用户查询、添加/删除好友、好友聊天、聊天记录管理、个人资料管理、查看好友资料、访问好友空间几大模块。如下图所示：

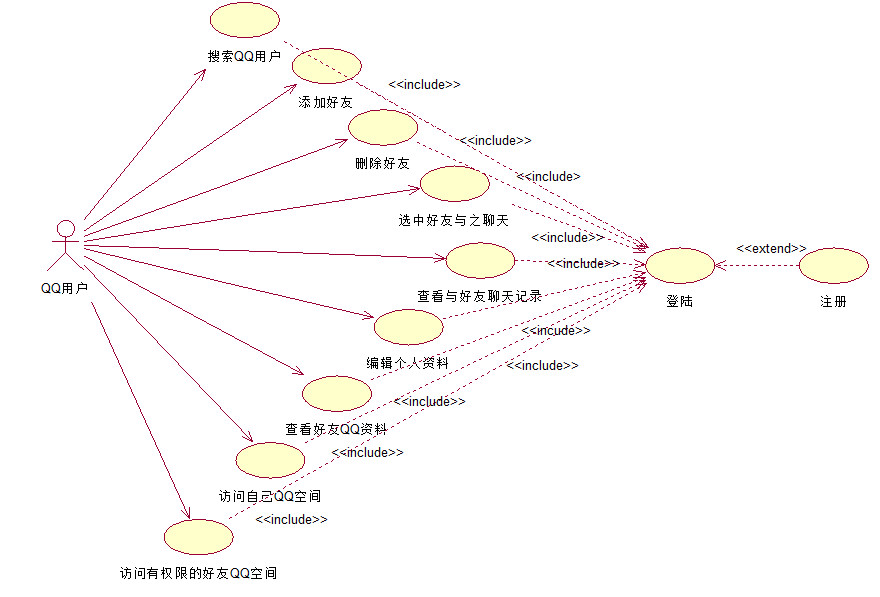


图3.3 QQ聊天软件用户功能用例图

QQ空间系统功能模块：主要分为系统登录退出、个人档管理(个人资料管理、空间访问权限管理)、日志管理、说说管理（说说评论管理）、相册管理、说说管理、留言管理几大模块。如下图所示：

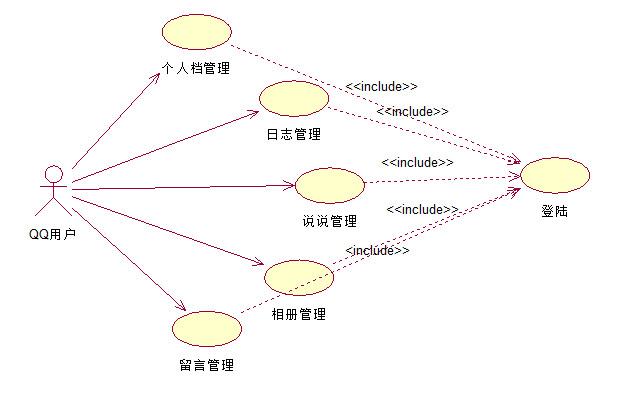


图3.4 QQ空间系统用户功能用例图

## 3.2 非功能性需求

### 3.2.1 安全型需求

安全性需求是软件设计中尤为重要的一个环节，安全性把控不好，将很可能会导致严重的后果，后期的维护和处理将会产生高昂的费用，甚至是不可挽回的。所以，对安全的把控和管理是必不可少的，本设计系统安全性从以下三个方面加以控制：

1：拦截QQ登陆和QQ空间登录访问的非法请求和非权限访问操作。

2：数据提交的校验需严格控制，除了前端校验外，服务后端也需提供数据格式、合法性检验。

3：保证数据的安全性，用户数据操作如出现异常，应由事务进行合理管理，从事务的传播行为和隔离级别等方面加以控制，以确保用户数据的安全性。

### 3.2.2 易用性和易维护性需求

本设计仿照腾讯QQ和QQ空间的UI进行设计，界面友好，具有良好的用户体验，用户可以像使用腾讯QQ和QQ空间一样来操作使用本系统，系统采用分层思想进行设计，仿QQ聊天软件的C/S程序采用界面和业务逻辑分离的Model2模式进行开发，仿QQ空间的B/S程序采用MVC模式进行开发，保证了可维护和可扩展性良好。

### 3.2.3 性能需求

数据查询模块：提交查询请求到取出数据响应到界面的时间不得超过5秒钟

数据持久化模块：用户提交数据管理请求响应处理完毕用时不得超过5秒钟。

服务响应模块：非操作数据库的服务请求响应用时不得超过3秒。

# 第四章 系统的设计与实现

## 4.1 系统总体设计

### 4.1.1 逻辑结构设计

仿QQ聊天软件子项目，采用C/S服务构架，整体采用Spring整合Mybatis，以Model2模式实现界面和业务逻辑的分离，界面开发采用基于Swing的GUI编程。服务端工程提供Dao层实现数据持久化，客户端工程通过Sockt编程发送消息包到服务端请求调用。

仿QQ空间系统子项目，采用B/S服务构架，整体框架采用Spring、SpringMVC、Mybatis整合来实现，采用MVC模式进行开发，SpringMVC充当控制器（Controller）、Mybaits完成数据持久化，服务端采用Jsp、Filte等技术实现和前端的交互，前端采用html5、响应式css框架Bookstrap、Javascript库Jquery，前后端数据交互采用Ajax和json。

MVC即Model（模型）、View（视图）、Controller（控制器），MVC模式是一种软件设计典范，用一种业务逻辑、数据和界面显示分离的方式组织代码，使得项目结构更加合理规范，大大增强了项目可扩展性，灵活性，降低了耦合度.

系统逻辑构架可参考图4.1：

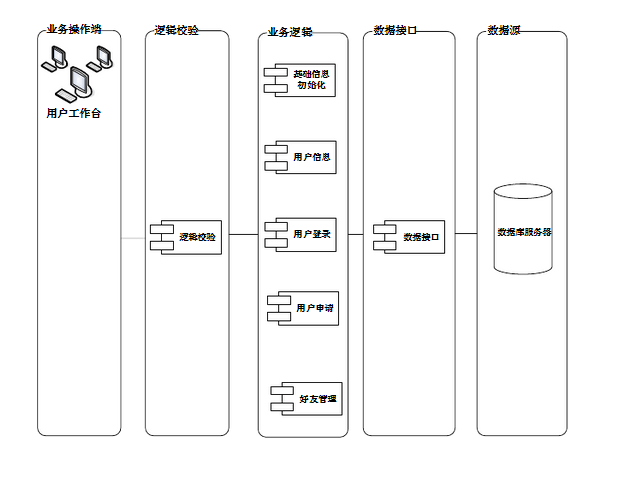


图4.1 系统逻辑架构图

工程结构分为View、Controller、Service、Dao层，具体阐述如下：

View：视图层，在仿QQ聊天软件子项目中，是基于swing的UI层；在仿QQ空间系统子项目中，是用于和前端交互的JSP来充当。

Controller：控制器层，在仿QQ聊天软件子项目中，自定义编写调用Service层的控制器类；在仿QQ空间系统子项目中，使用SpringMVC的Controller类来充当控制器。

Service：业务层，包含Model层，即数据模型（javabean），此层包含对业务逻辑的处理，通过调用Dao层封装的数据接口来实现数据持久化，同时提供给控制器层（Controller）调用。

DAO：数据持久层，编写一系列的数据操作接口，对数据进行增删改查操作，Dao层直接操作的对象是数据库。

系统工程结构图参考图4.2

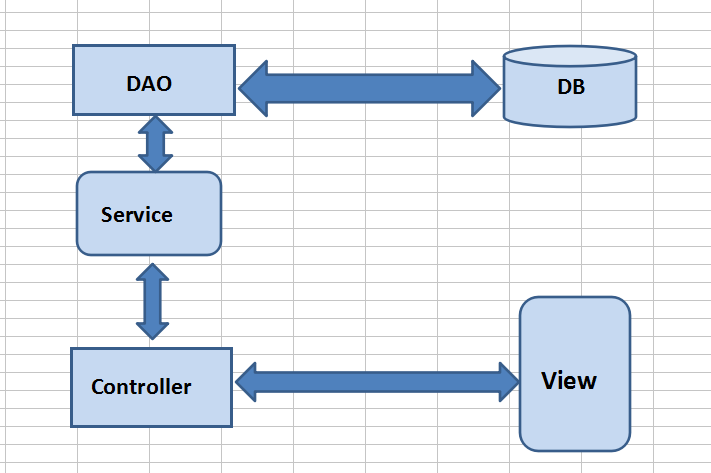


图4.2 系统工程结构图

### 4.1.2 开发环境

开发环境：JDk1.7+Eclipse+Myeclipse+Mysql6.0+Navicat+Tomcat7.0+DreamWeaver

硬件环境：CPU双核、频率1.7GHZ、运行内存4GB、硬盘500GB

软件环境：Windows7操作系统

## 4.2 基于UML的系统模型构建

通过UML建模分析，我们可以更加直观的得出系统从整体到细节的设计以及具体业务流程。

### 4.2.1 系统主要模型设计

1. 管理员登陆系统

事件流

当管理员登录时启动此用例。如下图所示：

基本流

1、用户输入用户名和密码后，系统判断其用户名和密码是否正确，如果正确，进入相应的系统操作界面：

2、如果登录失败，系统给出提示，登录重新开始：

3、退出登录，用例结束

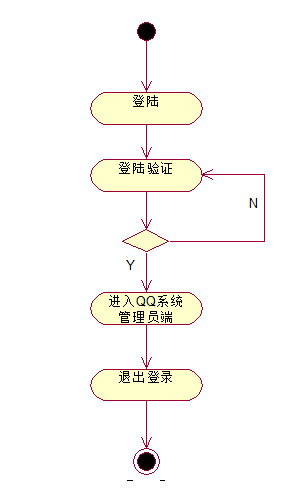


图4.3 管理员登陆过程活动图

（2）服务管理：

管理员登陆到系统后，可以手动启动服务和关闭服务，服务开启后，所有QQ用户方可登陆使用，关闭服务后，所有用户将被迫下线。如下图所示：

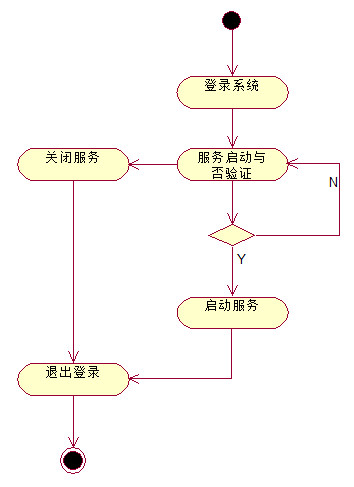


图4.4 用户管理活动图

（3）用户管理：

管理员登陆到系统后，可以查看用户列表、可以点击查看用户信息详情、可以强制指定的用户下线、冻结指定用户，被冻结的用户不能再登陆。如下图所示：

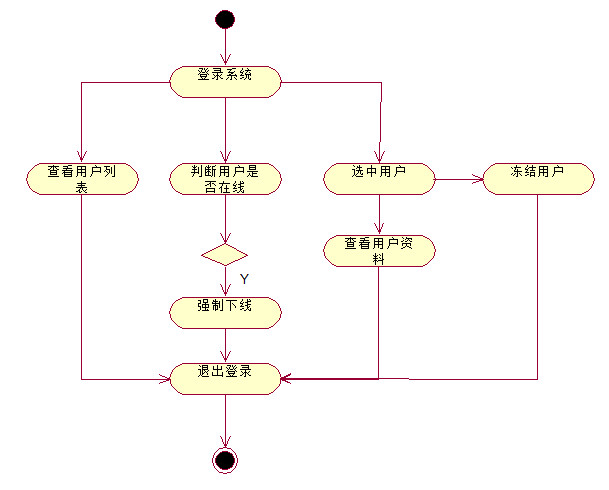


图4.5 服务管理活动图

1. QQ用户注册：

QQ用户注册，用户填写注册信息提交到服务器后，服务器后台会返回生成的QQ号码给用户。系统的QQ号码从10000开始产生，每次自增1。如下图所示：

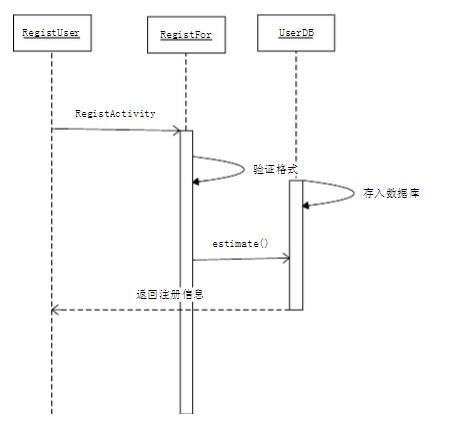


图4.6 用户注册动态模型

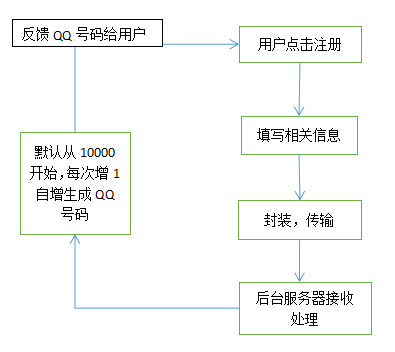


图4.7 用户注册流程图

（5）QQ用户登陆

用户登陆分为QQ登陆和QQ空间登录，QQ登陆和QQ空间登录数据验证分开验证，但是采用的是同一套数据库，QQ登陆成功后，也可以访问QQ空间。如下图所示：

事件流

当QQ用户登陆QQ或登陆QQ空间时启动此用例（前提是用户已经注册成功）

基本流

1、用户输入QQ账号和密码后，系统判断其用户名和密码是否正确，如果正确，进入相应的系统操作界面：

2、如果登录失败，系统给出提示，登录重新开始：

3、QQ登陆成功后，可以进入QQ系统操作界面，也可以通过QQ系统进入到QQ空间系统。

3、退出登录，用例结束

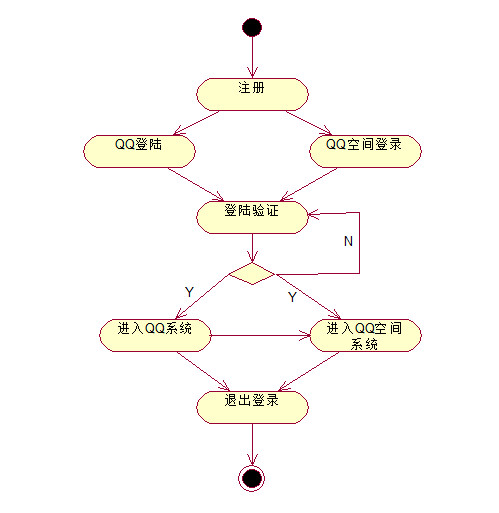


图4.8 QQ用户注册登陆过程活动图

（6）QQ好友管理：

QQ用户登陆到QQ系统后，进入到QQ面板界面，点击查询按钮，可以根据QQ号或昵称关键字查询QQ用户，可以选择查询结果并添加为QQ好友、可以从好友列表里面删除好友、可以选中好友查看个人信息、可以选中好友并访问好友空间，如下图所示：

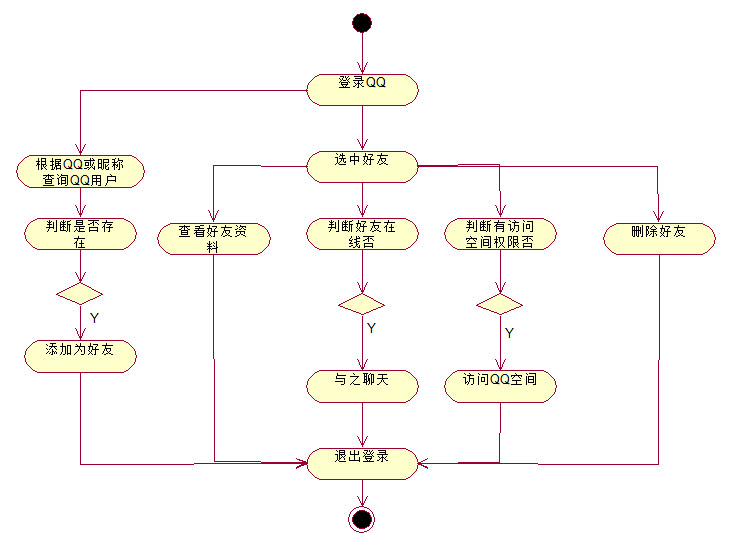


图4.9 QQ用户好友管理活动图

（7）QQ好友聊天管理：

QQ用户登录到QQ系统后，进入到QQ面板界面，从好友列表里面，选中一个在线好友标签并双击，打开聊天对话框可以与之聊天，也可以查看与该好友的聊天记录。如下图所示：

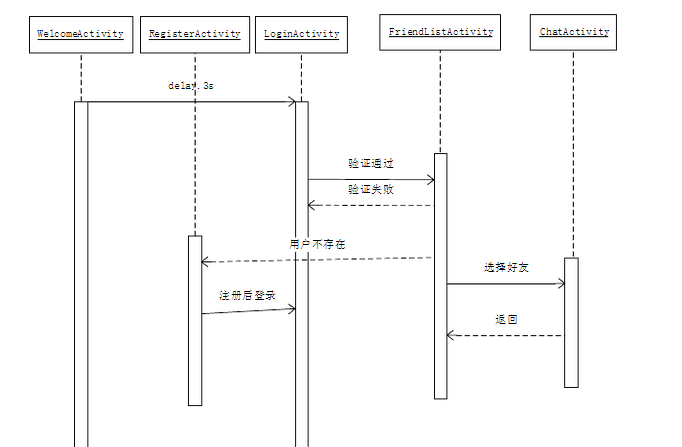


图4.10 系统动态模型

（8）QQ用户个人信息管理

QQ用户登录到系统后，进入到QQ面板界面，点击个人资料，查看个人信息，点击编辑资料按钮，进入到编辑界面，完成编辑并保存。

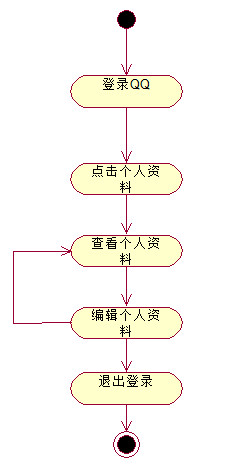


图4.11 个人信息管理活动图

（9）QQ空间个人档管理

QQ用户登录到系统后，进入QQ空间，点击个人档链接，可以设置空间访问权限（从QQ好友中筛选），可以查看和编辑个人资料。如果是访问的好友空间，只能查看好友个人资料。如下图所示：

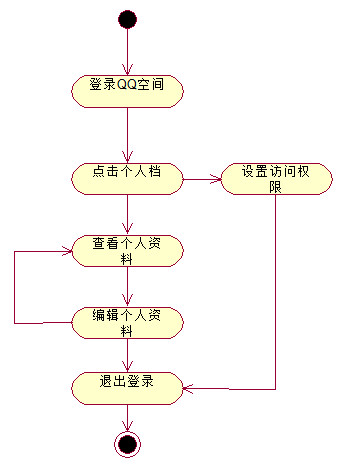


图4.12 QQ空间个人档管理活动图

（10）QQ空间日志管理

QQ用户登录到系统后，进入QQ空间，点击日志链接，可以分页查看日志列表，点击写日志，可以编写并发表日志，点击日志后面的阅览日志，可以进入该日志阅览页面进行查看，点击日志后面的删除链接，可以删除该日志。如果是访问好友空间，只能查看和阅览日志。如下图所示：

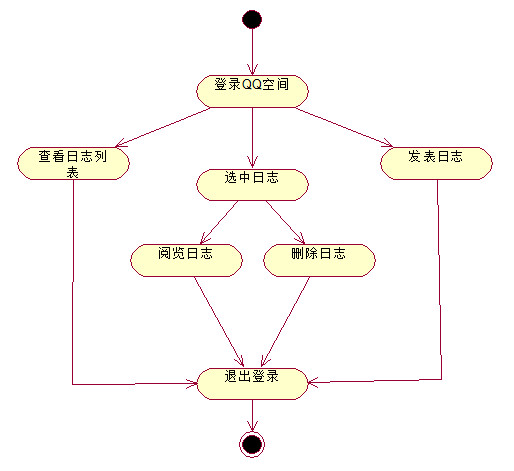


图4.13 QQ空间日志管理用例图

（11）QQ空间说说和评论管理

QQ用户登录到系统后，进入QQ空间，点击说说链接，可以查看说说列表（附带说说评论），点击发表说说，可以编写和发表说说，点击说说后面的评论，可以评论该说说，点击说说后面的删除，可以删除该说说，点击评论后面的删除，可以删除该评论。如果是访问好友空间，只能查看说说列表（附带说说评论）和发表评论。如下图所示：

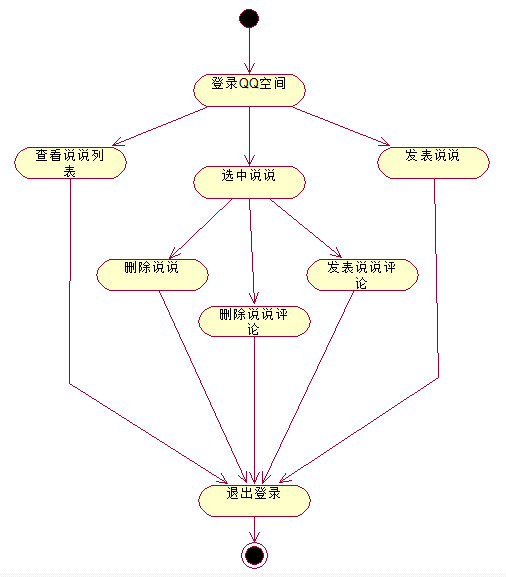


图4.14 QQ空间说说和评论管理活动图

1. QQ空间相册管理

QQ用户登录到系统后，进入QQ空间，点击相册链接，可以查看相册列表，点击新建相册，可以创建新相册；点击编辑，可以修改相册信息；点击删除，可以删除该相册；点击一个相册的封面即可进入照片列表，照片列表可以分页查看；点击上传照片，可以批量上传（每次最多5张）照片到该相册，并自动更新相册封面为最新照片；点击一张照片，可以浏览大图，点击删除，可以删除该照片，支持点击上一张/下一张浏览模式。如果是访问好友QQ空间，只能查看相册和阅览照片。如下图所示：

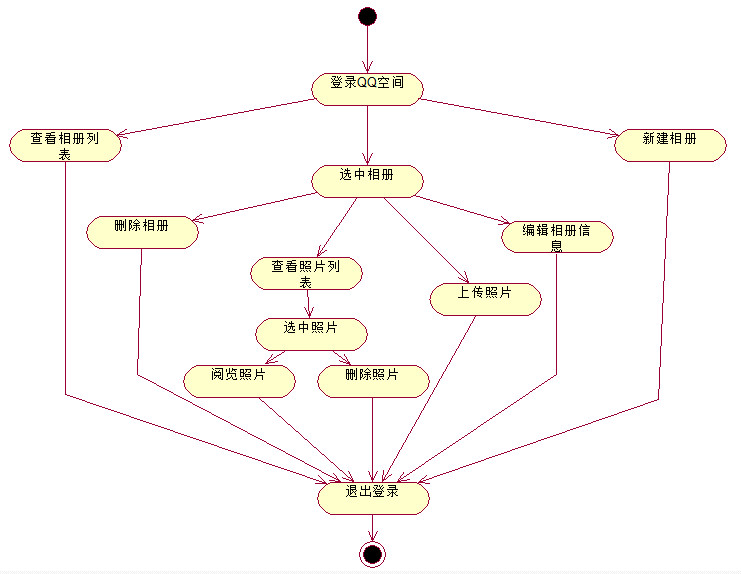


图4.15 QQ空间相册管理活动图

（13）QQ空间留言管理：

QQ用户登录到系统后，进入QQ空间，点击留言链接，可以分页查看留言列表；点击发表，可以编写并发表留言；点击留言后面的删除，可以删除该留言。如下图所示：

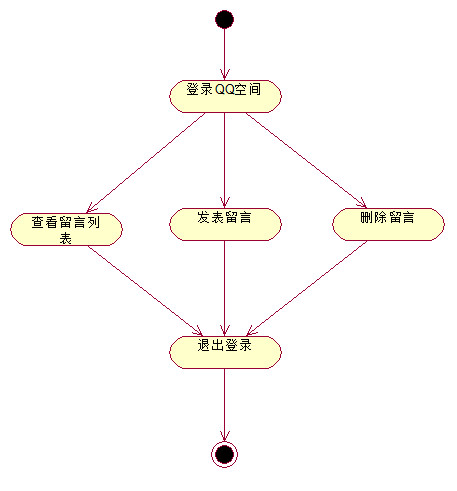


图4.16 QQ空间留言管理活动图

## 4.3 数据库表设计

本设计采用Mysql数据库，C3P0数据源，数据缓存采用Mybatis自带的缓存机制。

### 4.3.1 表结构和表间关系设计

本设计中仿QQ聊天软件和QQ空间系统共用一套数据库，经过详细的构思，设计出表清单如下：

表4.1 表清单

|  |  |
| --- | --- |
| 名称 | 注释 |
| QQADMIN | QQ管理员表 |
| QQUSER | QQ用户信息表 |
| QQ\_FRIEND | QQ好友表 |
| ZONEACCESSPERMISSION | 空间访问权限表 |
| DAILY | 空间日志表 |
| TALK | 空间说说表 |
| TALKCOMMENT | 说说评论表 |
| PHOTOALBUM | 空间相册表 |
| PHOTO | 空间照片表 |
| MESSAGE | 空间留言表 |

表间关系ER图：

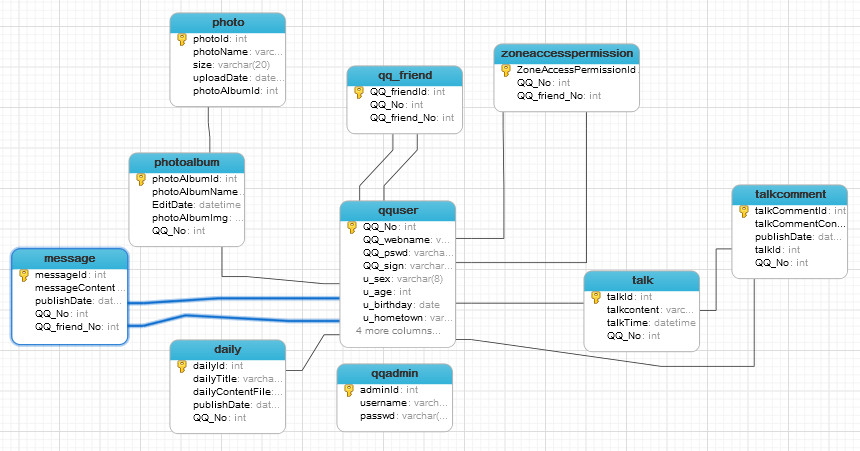


图4.17 表间关系图

1：QQ管理员表

QQ管理员表存储字段：主键ID、用户名和密码。

表4.2 QQADMIN表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |
| ADMINID | INT | 11 | 主键 | 否 | ID |

续表4.2 QQADMIN表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |
| USERNAME | VARCHAR | 20 |  | 否 | 用户名 |
| PASSWD | VARCHAR | 16 |  | 否 | 密码 |

2：QQ用户表

QQ用户表存储字段：主键QQ号码、网名（昵称）、QQ密码、个性签名、性别、年龄、生日、故乡、所在地（现居地）、职业、是否在线。其中主键QQ号从10000开始自增长产生，每次增加1。

表4.3 QQUSER表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |
| QQ\_NO | INT | 11 | 主键 | 否 | QQ号码 |
| QQ\_WEBNAME | VARCHAR | 30 |  | 否 | 网名(昵称) |
| QQ\_PSWD | VARCHAR | 20 |  | 否 | QQ密码 |
| QQ\_SIGN | VARCHAR | 50 |  |  | 个性签名 |
| U\_SEX | VARCHAR | 8 |  | 否 | 性别 |
| U\_AGE | VARCHAR | 4 |  |  | 年龄 |
| U\_BIRTHDAY | DATE |  |  |  | 生日 |
| U\_HOMETOWN | VARCHAR | 30 |  |  | 故乡 |
| U\_STAYCITY | VARCHAR | 30 |  |  | 所在地（现居地） |
| U\_EDUCATION | VARCHAR | 10 |  |  | 学历 |
| U\_PROFESSION | VARCHAR | 20 |  |  | 职业 |
| ISONLINE | INT | 4 |  |  | 是否在线 |

3：QQ好友表

QQ好友表存储字段：主键ID、本人QQ号码、好友QQ号码、其中主键ID号从1开始自增长产生，每次增加1。

表4.4 QQ\_FRIEND表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |
| QQ\_FRIENDID | INT | 11 | 主键 | 否 | ID |
| QQ\_NO | INT | 11 | 外键 | 否 | 本人QQ号 |
| QQ\_FRIEND\_NO | INT | 11 | 外键 | 否 | 好友QQ号 |

4：空间访问权限表

空间访问权限表存储字段：主键ID、本人QQ号码、好友QQ号码、其中主键ID号从1开始自增长产生，每次增加1。

表4.5 ZONEACCESSPERMISSION表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |

续表4.5 ZONEACCESSPERMISSION表续

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |
| ZONEACCESSPERMISSIONID | INT | 11 | 主键 | 否 | ID |
| QQ\_NO | INT | 11 | 外键 | 否 | 本人QQ号 |
| QQ\_FRIEND\_NO | INT | 11 | 外键 | 否 | 好友QQ号 |

5：空间说说表

空间说说表存储字段：主键ID、说说内容、发表时间、QQ号码，其中主键ID号从1开始自增长产生，每次增加1。

表4.6 TALK表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |
| TALKID | INT | 11 | 主键 | 否 | ID |
| TALKCONTENT | VARCHAR | 200 |  | 否 | 说说内容 |
| TALKTIME | DATETIME |  |  | 否 | 发表时间 |
| QQ\_NO | INT | 11 | 外键 | 否 | QQ号 |

6：空间说说评论表

空间说说评论表存储字段：主键ID、说说评论内容、发表时间、说说ID、QQ号码，其中主键ID号从1开始自增长产生，每次增加1。

表4.7 TALKCOMMENT表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |
| TALKCOMMENTID | INT | 11 | 主键 | 否 | ID |
| TALKCOMENTCONTENT | VARCHAR | 200 |  | 否 | 说说评论内容 |
| PUBLISHDATE | DATETIME |  |  | 否 | 发表时间 |
| TALKID | INT | 11 | 外键 | 否 | 说说ID |
| QQ\_NO |  |  | 外键 | 否 | QQ号 |

7：空间日志表

空间日志表存储字段：主键ID、日志标题、日志内容文件、发表时间、QQ号码，其中主键ID号从1开始自增长产生，每次增加1。

表4.8 DAILY表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |
| DAILYID | INT | 11 | 主键 | 否 | ID |

续表4.8 DAILY表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |
| DAILYTITLE | VARCHAR | 50 |  | 否 | 日志标题 |
| DAILYCONTENTFILE | VARCHAR | 50 |  | 否 | 日志内容文件 |
| PUBLISHDATE | DATETIME |  |  | 否 | 发表时间 |
| QQ\_NO | INT | 11 | 外键 | 否 | QQ号 |

8：空间留言表

空间留言表存储字段：主键ID、留言内容、发表日期、QQ号码、留言好友QQ号码，其中主键ID号从1开始自增长产生，每次增加1。

表4.9 MESSAGE表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |
| MESSAGE | INT | 11 | 主键 | 否 | ID |
| MESSAGECONTENT | VARCHAR | 200 |  | 否 | 留言内容 |
| PUBLISHDATE | DATETIME |  |  | 否 | 发表日期 |
| QQ\_NO | INT | 11 | 外键 | 否 | QQ号 |
| QQ\_FRIEND\_NO | INT | 11 | 外键 | 否 | 留言好友QQ号 |

9：空间相册表

空间相册表存储字段：主键ID、相册名称、更新时间、相册封面、QQ号码、其中主键ID号从1开始自增长产生，每次增加1。

表4.10 PHOTOAIBUM表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |
| PHOTOALBUMID | INT | 11 | 主键 | 否 | ID |
| PHOTOALBUMNAME | VARCHAR | 50 |  | 否 | 相册名称 |
| EDITDATE | DATETIME |  |  | 否 | 更新时间 |
| PHOTOALBUMIMG | VARCHAR |  |  |  | 相册封面 |
| QQ\_NO | INT | 11 | 外键 | 否 | QQ号 |

9：空间照片表

空间照片表存储字段：主键ID、照片文件名称、文件大小、上传时间、相册ID、其中主键ID号从1开始自增长产生，每次增加1。

表4.11 PHOTO表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段名 | 字段类型 | 长度 | 主键或外键 | 是否为空 | 描述 |
| PHOTOID | INT | 11 | 主键 | 否 | ID |
| PHOTONAME | VARCHAR | 50 |  | 否 | 照片文件名称 |
| SIZE | VARCHAR |  |  |  | 文件大小 |
| UPLOADDATE | DATETIME | 11 | 外键 | 否 | 上传日期 |
| PHOTOAIBUMID | INT |  |  |  | 相册ID |

### 4.3.2 数据连接池

本项目采用C3P0数据源，通过spring容器管理数据源datasource。

### 4.3.3 数据缓存

本项目数据持久层采用MyBatis框架，所以项目缓存采用MyBatis的缓存机制进行处理，缓存分为两个级别，即一级缓存和二级缓存

一级缓存是数据库Session级别的缓存，位于表示一次[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "http://blog.csdn.net/luanlouis/article/details/_blank)会话的SqlSession对象之中，又被称之为本地缓存。一级缓存在MyBatis中是内部实现的一个特性，不提供用户手动配置，默认情况下就自动支持一级缓存，用户没有定制它的权利（不过这也不是绝对的，可以通过开发插件对它进行修改）；二级缓存是Application应用级别的缓存，或者说是整个SessionFactory级别的缓存，它的是生命周期很长，跟Application的生命周期一样，也就是说它的作用范围是整个Application应用。

 MyBatis中一级缓存和二级缓存的组织如下图所示：

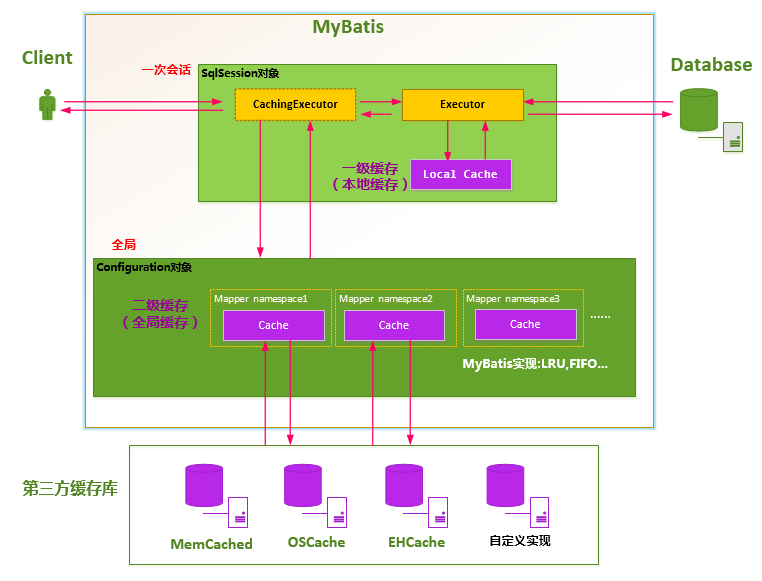


图4.18 Mybatis缓存架构图

一级缓存与二级缓存可以单独使用，也可以组合使用，当组合使用时，所有的数据查询请求，首先到本地缓存中查找，如果有直接将数据返回，如果没有则从二级缓存中检索，如果找到则将数据缓存到一级缓存，同时将数据返回；如果一、二级缓存中都不存在该数据，则从数据库中查询，将查找出来的结果，同时维护到一级与二级缓存中。

缓存回收策略：

LRU（最近最少使用），缓存框架为该算法维护一个缓存项队列，队列中的缓存项按照每项的最后被访问时间排序。在回收时会删除最后一次被访问时间距离现在最久的项。

LFU（最不经常使用），缓存框架为每个缓存对象计算他们被使用的频率，在回收时缓存空间会把最不常用的缓存对象销毁。

FIFO（先进先出），缓存框架会对缓存的对象按照加载的先后顺序来决定是否回收，在回收时先回收最早加载的缓存对象占用的空间。

## 4.4 系统主要功能模块的设计与实现

本设计对应两个子项目，即仿QQ聊天软件项目和QQ空间系统项目，工程名分别为：QQ\_Server、QQ\_Client、QQ\_Zone，其中QQ\_Server和QQ\_Client分别为仿QQ聊天软件子项目的服务端工程和客户端工程；QQ\_Zone为web项目，即QQ空间系统工程。

### 4.4.1 QQ用户注册与登录

QQ注册功能位于QQ\_Zone工程中，QQ注册页面可以通过QQ登陆界面链接访问，也可以通过QQ空间登录页面链接访问，前端提交form注册信息表单到控制器Controller，QQ号码从1000开始生成，每次自增1，所以需要先获取数据库中用户当前QQ号码的最大值，如果返回为空说明当前还没有用户注册，则生成的QQ号码是10000，如果不为空，则生成的QQ号码加1。用户申请成功后，系统将生成的QQ号通过AJAX反馈给用户。



图4.19 QQ账号注册界面

详细的实现过程请参考项目源码。

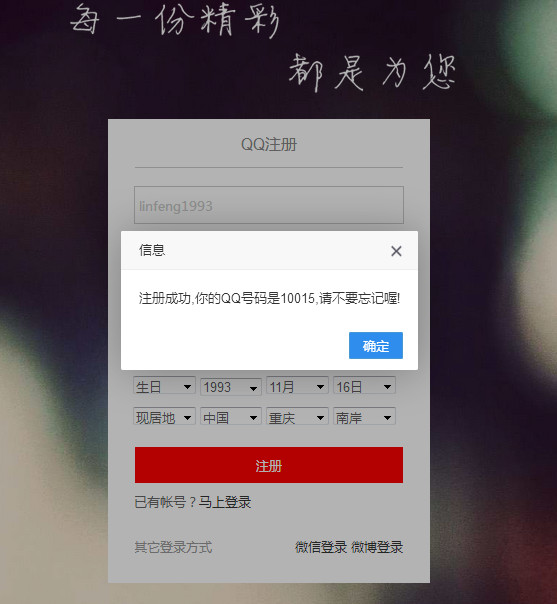


图4.20 QQ账号注册成功提示界面

由于在仿QQ聊天软件子项目中，DAO层位于服务端工程，即QQ\_Server工程，所以，来自客户端工程的所有请求，包括用户登录都需要通过Socket通信发送消息包到服务端工程进行中转处理，不同的请求与不同的反馈结果都有着我们自定义的类型，于是我们封装一个消息包类并定义各种消息包的类型，然后在服务器端工程和客户端工程中同时编写一个信息包类，里面可以定义如消息类型、发件人，发件内容，收件人等属性，这样服务器端和客户端通信线程每次都通过消息包来传递信息。这样服务器和客户端就可以进行自由的交互了。启动QQ服务器（实例化一个服务器类对象）是前提。

由于客户端机器上可能同时登陆多个QQ号码，也有可能服务器同时会接收到多个客户端用户发来的不同请求，于是我们不得不考虑线程并发问题，我们把服务器端的Socket通信做成一个线程类，当接收到一个客户端用户发来的正确的登陆请求完成登陆后，即在服务端开启一个专门与该用户通信的线程，这样该用户后续发来的请求都会直接与该线程完成Socket数据交互；同理，在客户端，当用户接收到来自服务端反馈的自己登陆成功的消息后，即在客户端本地开启一个线程用于专门管理自己的通信，这样就解决了本地存在多个QQ用户同时登陆使用时导致的线程紊乱问题。

另外值得说明的是，如何从QQ面板进入到注册页面呢？我的处理方式是在QQ登录面的注册账号标签上注册一个点击事件， 然后通过Runtime.getRuntime().exec()方法将访问请求发送到服务器（请求映射QQ账号注册页面的路径）前提是QQ空间系统服务已经启动，因为QQ注册页面在QQ空间项目中。

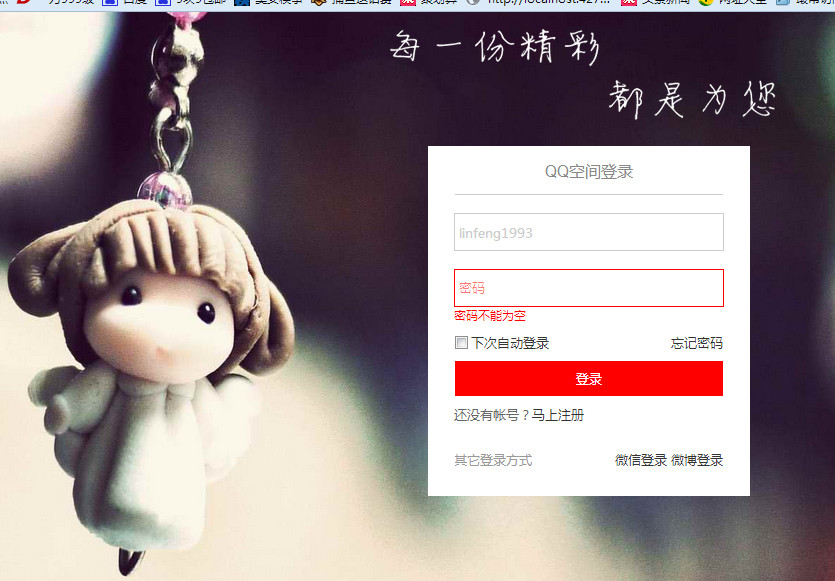


图4.21 QQ空间登陆界面



图4.22 QQ登录界面

详细的实现过程请参考项目源码。

### 4.4.2 QQ好友添加与删除

在用户已经登陆成功的基础上，QQ好友的添加和删除就变得尤为简单了，我们已经规定了每种消息包的种类，于是，只需改变客户端发送请求和服务端反馈结果的消息种类。

另一方面，我们应该对QQ的面板进行一个管理，考虑到如果本地有多个QQ账号登录，那么QQ好友添加和删除操作成功后，好友列表需要及时刷新，如果对每个登录成功的账号的QQ面板不做区分的话，那么面板刷新会出现线程紊乱。所以，在登录成功之后，我们需要将每个QQ账号的面板存入一个HasMap里面，键由QQ号充当，这样的话，当添加或删除好友成功后收到服务器的反馈时，能够正确定位并刷新相应的好友QQ面板。

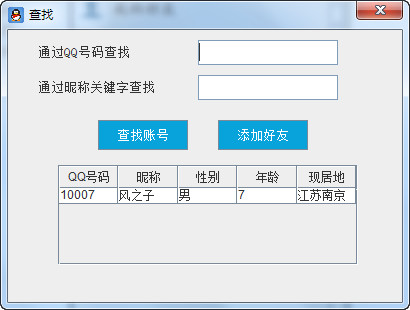


图4.23 QQ用户查找界面

详细的实现过程请参考项目源码。

### 4.4.3 QQ好友上下线提示

同理于QQ用户登录和好友的添加与删除，QQ用户登陆成功后，其所有在线好友的QQ面板上的好友列表将会实时刷新，上线好友头像变成彩色，反之，头像变成灰色。我们规定好QQ好友上线和下线提示的包即可。

****

图4.24 QQ面板好友上线界面

详细的实现过程请参考项目源码。

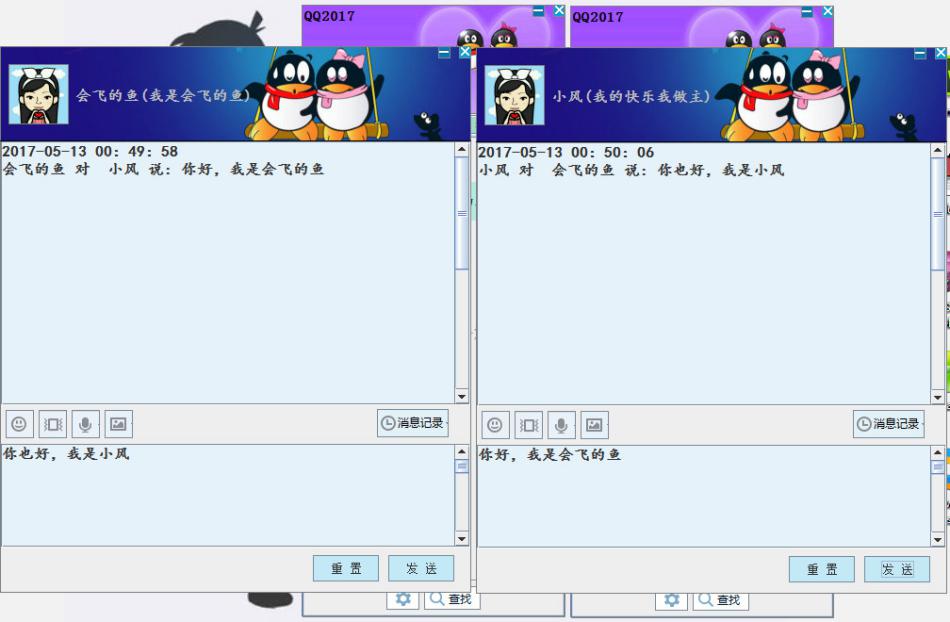
### 4.4.4 QQ好友多对多聊天

基于前面展示的一系列的功能，我们可以轻易实现该功能，当客户端用户将聊天信息发送给好友时，先是将聊天信息封装成信息包发送到服务器，服务器再解析出该信息包中的收信人，最后将信息包的内容转发给收信人。

这里，也需要特别说明的是，为了避免服务器将聊天内容转发后，客户端聊天对话框出现线程紊乱的情况，我们每次在双击点开一个好友对话框后就将该对话框存入一个HashMap中，键由本人QQ和好友QQ拼接而成，保证了其唯一性。这样，当客户端线程接收到服务器反馈回的聊天记录后，能保证聊天内容能显示到指定的聊天对话框上。

只有这样处理，那么才能保证在本地登陆多个QQ账号后，可以实现好友多对多的聊天，

其实，如果本设计需要实现群聊功能的话，那么也必须按照这个思路来设计，毕竟，任何两个QQ账号之间的聊天都需要通过一个独立的通信线程。

**** 图4.25 QQ好友聊天界面

需要说明的是，由于QQ好友的添加与删除以及好友聊天当中涉及到的类只有一个QQUser和QQFriend表，其类图可以表示为下图所示：

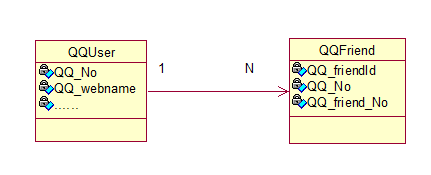


图4.26 QQ用户和好友关系类图

详细的实现过程请参考项目源码。

### 4.4.5 QQ个人信息管理

由于个人信息管理模块相对简单，在前面几个功能模块的基础上，这里我不再赘述。

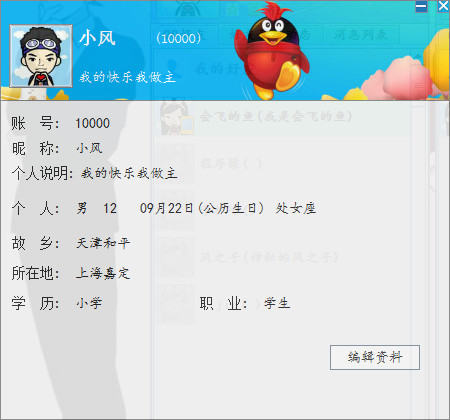
****

图4.27 QQ个人资料界面

### 4.4.6 QQ空间说说评论管理

QQ空间说说以及说说评论模块的功能是系统中一个常见的增删改查操作。这里我们具体说明一下模块的功能:

QQ说说评论是关联QQ说说的，所以在说说主页是能按照时间由近及远的顺序展示说说列表，每条说说的评论显示在说说下面，我们可以删除一条具体的评论，也可以在指定的说说后面发表评论，当然也可以删除指定的说说，删除说说其下的所有评论会一并删除。最后需要说明的是，如果是访问好友是QQ空间的话，是不能发表说说的。

****

图4.28 QQ空间说说主页

详细的实现过程可以参考项目源码。

### 4.4.7 QQ空间相册管理

QQ空间相册模块的设计和实现上其实有可圈可点的地方，但基本的增删改查操作在这里我不再赘述。功能模块上有几点需要说明:

1：上传照片支持批量上传，但限制每次最多上传5张照片。

2：新建相册后，相册的封面是没有图片的，但上传照片到该相册后，其封面图片自动更新为最后上传到该相册的照片。

3：阅览照片大图的时候，支持上一张/下一张阅览模式。

4：相册不支持分页浏览，但是照片支持分页查看，照片列表中的每一张照片小图并不是按照一定比例缩小显示，而是使用CSS进行溢出隐藏控制，照片只显示固定的区域大小，而点击照片小图后照片大图按照原图的长宽80%显示在一个弹框中。弹框右上角提供了删除该照片的超链接，点击删除会删除该照片并返回此页照片列表。弹框左下角和右下角分别有上一张和下一张的超链接，当点击上一张或下一张后，显示对应的照片，但是此时不提供删除该照片的功能。

****

图4.29 QQ空间相册主页

我重点说明以下模块：

照片阅览时支持上一张/下一张浏览模式，这里我的设计思路是，在页面上，点击一张照片链接后触发一个javascript函数，将该照片的信息作为参数传给该javascript函数，然后弹出一个默认隐藏的div窗口，让照片显示在上面，点击上一张或下一张的时候，将该照片的主键ID，所属相册ID信息通过AJAX传到服务端后台，到数据库中去查找匹配，获取并返回上一张或下一张照片的信息，通过json格式化后传回页面，最后javascript解析数据并更改当前弹框div显示的照片即可。

**** 图4.30 QQ空间相册照片列表

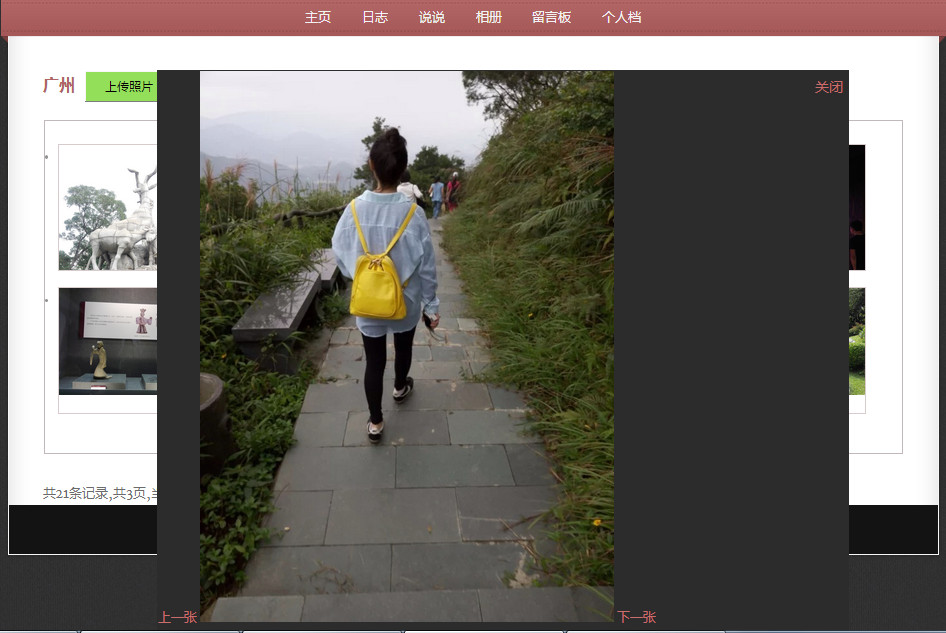
****

图4.31 QQ空间相册照片阅览

详细的实现过程请参考项目源码。

### 4.4.8 QQ空间留言管理

由于留言板模块可以看似是一个缩减版的说说模块，但是留言板的不同在于不管是访问自己的QQ空间还是访问好友的QQ空间，都是可以留言的。具体实现可以参考项目源代码。

**** 图4.32 QQ空间留言板主页

详细的实现过程请参考项目源码。

# 第五章 系统测试

## 5.1 系统测试用例设计

软件测试是一门通过输入输出的实际结果和期望结果来描述软件功能实现的正确与否的学科，其目的是确定应用程序的某个特性是否正常的工作。软件测试用例的基本要素包括测试用例编号、测试标题、重要级别、测试输入、操作步骤、预期结果。

下面我将对本设计进行单元测试和集成测试。

### 5.1.1 单元测试

模块测试是针对概要设计中的每一个模块来进行测试的，目的是保证每个模块作为一个单元能够正常运行，所以模块测试通常被称为单元测试。

下面我对本设计的主要模块进行单元测试，用户注册登录模块结果如下表:

表5.1 用户注册登录模块单元测试报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试人： | 李林峰 | 测试时间： | 2017/5/10 | 错误个数： | 0 |
| 序号 | 功能模块 | | 输入 | 输出 | 实际结果 |
| 1 | QQ注册 | | 访问注册页面，不合法填写注册信息/合法填写注册信息 | 提示输入不合法/注册成功，成功返回申请的QQ号码 | 提示输入不合法/注册成功，成功返回申请的QQ号码 |
| 2 | QQ登陆 | | 输入错误的账号和密码/正确输入账号和密码 | 提示账号或密码错误/登陆成功，进入QQ面板并正确显示好友列表信息 | 提示账号或密码错误/登陆成功，进入QQ面板并正确显示好友列表信息 |
| 3 | QQ空间登录 | | 输入错误的账号和密码/正确输入账号和密码 | 不能进入QQ空间/能成功进入QQ空间主页 | 不能进入QQ空间/能成功进入QQ空间主页 |

好友管理模块的单元测试结果如下表:

表5.2 好友管理模块单元测试报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试人： | 李林峰 | 测试时间： | 2017/5/10 | 错误个数： | 0 |
| 序号 | 功能模块 | | 输入 | 输出 | 实际结果 |
| 1 | 根据QQ号查询QQ用户 | | 在QQ面板上点击查询按钮进入查询页面，输入QQ号点击查询 | 返回匹配成功的QQ号信息，若不存在则显示空 | 返回匹配成功的QQ号信息，若不存在则显示空 |

续表5.2 好友管理模块单元测试报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试人： | 李林峰 | 测试时间： | 2017/5/10 | 错误个数： | 0 |
| 序号 | 功能模块 | | 输入 | 输出 | 实际结果 |
| 2 | 根据昵称关键字查询QQ用户 | | 在QQ面板上点击查询按钮进入查询页面，输入昵称关键字点击查询 | 返回匹配成功的QQ号信息，若不存在则显示空 | 返回匹配成功的QQ号信息，若不存在则显示空 |
| 3 | 添加好友 | | 选中查询返回的QQ用户，点击添加好友 | 如果选中的用户和自己QQ号一致，提示不能添加自己为好友；如果选中的用户和自己的某位QQ好友一致，则提示该用户已经是你的好友了，不需要重复添加；否则，提示添加确定与否,点击确定即可添加为好友，并刷新好友面板，点取消则不添加 | 如果选中的用户和自己QQ号一致，提示不能添加自己为好友；如果选中的用户和自己的某位QQ好友一致，则提示该用户已经是你的好友了，不需要重复添加；否则，提示添加确定与否,点击确定即可添加为好友，并刷新好友面板，点取消则不添加 |
| 4 | 删除好友 | | 在QQ面板好友列表上右键点击好友标签选中删除一项，然后点击 | 提示确定要删除该好友，点确定即可删除并刷新好友列表；点取消则不删除该好友 | 提示确定要删除该好友，点确定即可删除，并刷新好友列表；点取消则不删除该好友 |
| 5 | 编辑个人资料 | | 在QQ面板上点击个人资料图标 | 弹出个人信息面板，点击编辑资料，进入编辑界面，完成编辑点保存提示编辑成功 | 弹出个人信息面板，点击编辑资料，进入编辑界面，完成编辑后点保存提示编辑成功 |
| 6 | 查看好友资料 | | 在QQ面板好友列表上右键点击好友标签选中查看好友资料一项 | 弹出该好友的个人信息面板，并正确显示信息 | 弹出该好友的个人信息面板,并正确显示信息 |

QQ好友聊天和QQ空间模块单元测试结果如下表：

表5.3 QQ好友聊天和QQ空间模块单元测试报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试人： | 李林峰 | 测试时间： | 2017/5/10 | 错误个数： | 0 |
| 序号 | 功能模块 | | 输入 | 输出 | 实际结果 |
| 1 | 访问QQ空间 | | 在QQ面板上点击QQ空间图标/在QQ面板好友列表上右键点击好友标签选中访问QQ空间一项，然后点击 | 进入自己的QQ空间主页/如无访问权限，提示无访问权限，否则进入到该好友QQ空间 | 进入自己的QQ空间主页/如无访问权限，提示无访问权限，否则进入到该好友QQ空间 |
| 2 | 选择好友打开对话框聊天 | | 在QQ面板好友列表上双击选中好友标签 | 弹出与该好友聊天的对话框，如果对方在线，可以和他聊天 | 弹出与该好友聊天的对话框，如果对方在线，可以和他聊天 |
| 3 | 登陆3个QQ用户两两互聊 | | 登陆3个QQ用户，保证每个账号都有两个以上好友，然后让每个用户都打开两个好友对话框，进行聊天 | 聊天能正常进行，数据不会紊乱 | 聊天能正常进行，数据不会紊乱 |
| 4 | 查看说说列表和评论 | | 点击QQ空间的说说链接 | 显示该QQ空间主人的所有说说列表和评论 | 显示该QQ空间主人的所有说说列表和评论 |
| 5 | 发表说说 | | 在说说编辑框里面编写完内容，然后点击发表说说按钮/点击放弃按钮 | 如果编辑框内容为空，提示输入不能为空，否则发表说说成功，并刷新说说列表/关闭编辑框，不发表 | 如果编辑框内容为空或字数上限，则给出相应的提示，否则发表说说成功，并刷新说说列表/关闭编辑框，不发表 |

续表5.3 QQ好友聊天和QQ空间模块单元测试报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试人： | 李林峰 | 测试时间： | 2017/5/10 | 错误个数： | 0 |
| 序号 | 功能模块 | | 输入 | 输出 | 实际结果 |
| 6 | 发表说说评论 | | 点击说说后面的评论链接 | 弹出评论框，输入评论内容，点击评论按钮/点击放弃按钮 | 如果评论内容为空或字数上限，则给出相应的提示，否则发表评论成功，并刷新说说评论列表/关闭编辑框，不评论 |
| 7 | 删除说说和评论 | | 点击说说后面的删除链接/点击说说评论后面的  删除链接 | 提示确定要删除吗，点确定即删除该说说说并刷新说说列表，点取消则不删除/提示确定要删除吗，点确定即删除该评论并刷新说说评论列表，点取消则不删除 | 提示确定要删除吗，点确定即删除该说说说并刷新说说列表，点取消则不删除/提示确定要删除吗，点确定即删除该评论并刷新说说评论列表，点取消则不删除 |
| 8 | 查看相册列表 | | 点击QQ空间里的相册链接 | 显示该QQ空间主人的相册列表 | 显示该QQ空间主人的相册列表 |
| 9 | 编辑相册 | | 点击相册下方的编辑按钮 | 弹出编辑框，更改相册名称，点编辑便更改成功并刷新相册列表，点放弃则关闭编辑框，不更改 | 弹出编辑框，更改相册名称，点编辑便更改成功并刷新相册列表，点放弃则关闭编辑框，不更改 |
| 10 | 删除相册 | | 点击相册下方的删除链接 | 提示确定要删除该相册，点确定即删该相册并刷新相册列表/点取消则不删除 | 提示确定要删除该相册，点确定即删该相册并刷新相册列表/点取消则不删除 |

续表5.3 QQ好友聊天和QQ空间模块单元测试报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试人： | 李林峰 | 测试时间： | 2017/5/10 | 错误个数： | 0 |
| 序号 | 功能模块 | | 输入 | 输出 | 实际结果 |
| 11 | 新建相册 | | 点击新建相册按钮 | 弹出编辑框，填写相册信息点立即新建按钮，如果相册名为空则提示相册名不能为空，否则新建成功并刷新相册列表/点击放弃按钮，关闭编辑框，不创建 | 弹出编辑框，填写相册信息点立即新建按钮，如果相册名为空则提示相册名不能为空，否则新建成功并刷新相册列表/点击放弃按钮，关闭编辑框，不创建 |
| 12 | 分页查看照片列表 | | 点击一个相册的封面连接 | 进入该相册照片列表的第一页，如果有上一页或下一页，点击可以分页查看 | 进入该相册照片列表的第一页，如果有上一页或下一页，点击可以分页查看 |
| 13 | 阅览照片大图 | | 点击一张照片链接 | 弹出框显示照片大图，左下角有上一页链接，右下角有下一页连接，右上角有删除和关闭链接，点关闭链接则关闭弹框 | 弹出框显示照片大图，左下角有上一页链接，右下角有下一页链接，右上角有删除和关闭链接，点关闭链接则关闭弹框 |
| 14 | 阅览上一张照片 | | 点击上一张链接 | 弹框显示的照片切换到上一张，如果已经是第一张了，则提示没有了 | 弹框显示的照片切换到上一张，如果已经是第一张了，则提示没有 |
| 15 | 阅览下一张照片 | | 点击下一张链接 | 弹框显示的照片切换到下一张，如果已经是最后一张了则提示没有了 | 弹框显示的照片切换到下一张，如果已经是最后一张了，则提示没有了 |

续表5.3 QQ好友聊天和QQ空间模块单元测试报告

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 测试人： | 李林峰 | 测试时间： | 2017/5/10 | 错误个数： | 0 |
| 序号 | 功能模块 | | 输入 | 输出 | 实际结果 |
| 16 | 删除照片 | | 点击删除链接 | 提示确定要删除，点确定即删除该照片并刷新照片列表，点取消则不删除 | 提示确定要删除，点确定即删除该照片并刷新照片列表，点取消则不删除 |
| 17 | 分页查看留言列表 | | 点击QQ空间的留言板链接 | 分页显示该QQ空间主人的所有留言列表 | 分页显示该QQ空间主人的所有留言列表 |
| 18 | 发表留言 | | 在编辑框填写留言内容后点击发表留言按钮/点击放弃按钮 | 如果留言内容为空，则提示内容不能为空，否则发表成功并刷新留言列表/不发表 | 如果留言内容为空或字数上限，则给出相应的提示，否则发表成功并刷新留言列表/不发表 |
| 19 | 删除留言 | | 点击留言后面的删除链接 | 提示确定要删除该留言，点确定即删除该留言并刷新留言列表，点取消则不删除 | 提示确定要删除该留言，点确定即删除该留言并刷新留言列表，点取消则不删除 |

### 5.1.2 集成测试

集成测试是在单元测试的基础上，将所有模块按照设计要求（如根据结构图）组装成为子系统或系统，进行集成测试。

实践表明，一些模块虽然能够单独地工作，但并不能保证连接起来也能正常的工作。一些局部反映不出来的问题，在全局上很可能暴露出来。

在上面单元测试的基础上，下面我们再对本设计进行系统集成测试，结果如下表：

表5.4 集成测试报告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 集成测试范围 | 每一个模块联合在一起测试 | | | |
| 集成测试方法 | 人工进行输入内容 | | | |
| 集成测试的环境 | Eclipse、火狐浏览器、IE浏览器、Mysql | | | |
| 集成测试任务 | 需测试的功能 | 预测结果 | 进度日期 | 实际日期 |
| 各模块之间组装测试 | 各个部分之间传递的输入对象是否正确 | 通过 | 2017/5/10 | 2017/5/10 |

续表5.4 集成测试报告

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 各个界面的切换 | 点击相对应的功能能否切换为正确的界面 | 通过 | 2017/5/10 | 2017/5/10 |

# 第六章 总结和展望

仿QQ聊天软件和QQ空间系统相对真实的腾讯QQ和QQ空间来说，要显得简单和小巧得多。但是最基本的功能都已经具有了，可以当做一个简化版QQ和QQ空间来使用，但是由于时间问题和个人能力的不足，某些功能还没有完全实现，比如QQ头像，由于QQ是C/S构架程序，而QQ空间是Web程序，要想让QQ头像在两个子项目中共享，web服务器下和本地服务器（本机）上如何避免QQ头像文件分两地存储，这尚需进一步优化完善，当然还有更多腾讯QQ和QQ空间的功能不在本次设计范围之内，这也算是留给在往后的日子里的一个拓展的机会吧。

本设计重点运用了基于TCP/IP的Socket编程，由于程序中处处可能涉及到线程并发的问题，所以在编写的过程中，收获还是很大的，设计期间，也针对Java多线程并发编程进行了学习补充。本设计两个子项目均使用了Spring和Mybatis两大框架，之所以这样选择，是从安全性、高效性等多方面进行了考虑，如今日益流行的框架越来越多，框架的底层实现原理虽然很简单，java类加载、类反射、自定义注解结合dom解析基本也是实现的一种套路。很多人都能写出自己设计的框架，但是真正被大众认可的却少之又少，框架设计的成功与否取决于设计的思想的高度，设计模式也很重要。

经过这次毕业设计，我最大的收获是有了程序跨域和跨平台的思想转变，web应用和桌面程序如何交互，后续我会考虑到如何使用JNI去实现跨语言的调用。当前目前我的认知是严重不足的，但是这可以作为往后对自己的一个期望和计划。

**致谢**

工作确实一度繁忙，当初开题的时候题目定的有点大，QQ集成QQ空间和邮箱系统，以前有使用过很多次JavaMail发送邮件，但是没考虑到接收邮件到自己编写的web邮箱系统是一个很大的问题。目前也不在我所能及的范围内，非常感谢石晔琼老师的建议，我删除掉了邮箱系统这个模块，同时在排版和内容编排上，石老师也给予了建设性的意见，让整篇论文从理论到具体实现条例更加清晰。

同样感谢公司同事在Socket编程实战方面的指导，同事推荐了设计模式和并发编程方面的相关书籍。通过这次毕业设计，我认识到了自己在设计模式和如何优化项目方面的严重不足，在未来的学习过程中，我将会把更多的经历投入到系统的结构和优化上，以及安全和性能的把控上，以便以后在接触到大型项目的开发时能够得心应手。

最后衷心感谢学校和学院的培养，我不但学到了很多计算机专业的理论知识，同时也具备了软件开发的基本素养和能力。

**参考文献**：

[1] 邢东旭,王峰;基于Spring框架的Web系统的研究与应用[J];内蒙古科技大学学报;2012年04期

[2] 刘高军;夏景隆;基于Spring MVC和iBATIS框架的研究与应用[J];计算机安全;2012年07期

[3] (美)威尔德，斯尼德等著，赵利通译. SpringFramework 2入门经典[M]. 北 京：清 华大学出版社，2009

[4] 陈雪莲. 网络与 Web 技术导论[T].北京：清华大学出版社，2009:10-20.

[5] 闻力欧. 网页制作实用教程[T].北京：清华大学出版设，2009:10-24.

[6] 张桂珠. Java 面向对象程序设计[T].第三版.北京.清华大学出版社，2010.

[7] Spring与SpringMVC的容器关系分析 - OPEN 开发经验库

[8] Craig Walls, Ryan BreidenBach. Spring In Action[M]. Manning Publications. 2006

[9] 何玉洁. 数据库原理与应用教程[M]. 机械工业出版社，2005.12

[10] 刘玉平. Java输入输出流及在网站开发中的应用探讨[J]. 郴州市科技局，2009

[11] 邝孔无，王晓敏. 信息系统分析与设计(第三版)[M]. 清华大学出版社，2006：23-26.

[12] 张虹. 软件工程与软件开发工具．北京：清华大学出版社，2009：55-75

[13] Bruce Eckel. Thinking in Java．北京：机械工业出版社，2009：42-64

[14] 罗军舟等. TCP/IP协议及网络编程技术．北京：清华大学出版社，2008：132-210

[15] 孙一林，彭波. Java网络编程实例．北京：清华大学出版社．2003：38-65

[16] 赵海亮. 局域网聊天程序的分析与实现[J].北京:中国科技信息.2004,(22):17-18.

[17] 马勇. 基于UDP多播文件传输方法的研究[J].福州:福建电脑.2009,(2):69-70.

**Java language development like QQ chat software**

**LI Lin-feng**

**Abstract**: This paper is based on Java -like QQ chat software, QQ Zone systems, simulation Tencent QQ chat software features, such as registration, login, add friends, friends chat, file-transfer function such as, followed by simulated QQZone frequently used functions, such as publishing say, upload photo album, journal and other functions, to a better user experience for the purpose of developing a simple interface, instant messaging chat feature is available interactive software.

This paper first describes the background of the chat software, followed by the purpose and significance of the project, and then introduces software development environments and development tools and technologies, gives the software system architecture, flow charts of each module and use cases, and then detail the design and implementation of software, and finally, the system was simple test.

**Key words:** Java; QQ chat software; QQ Zone: user experience