

Instant Messaging System Design and Development for Enterprise Internal Information Exchanges Based on Android Platform

Xinlu Du, Xingyu Wang

North China University of Technology, Beijing
Email: duxinlu81@live.com

Received: Nov. 28th, 2016; accepted: Dec. 16th, 2016; published: Dec. 19th, 2016

Copyright © 2016 by authors and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

To boost efficiency of the employees who have various project assignments, it is necessary to develop a proper information exchanging software. The paper presents an IMS (instant messaging system) for employees who belong to the same particular enterprise, with functionalities including: chitchat, file transform and resources sharing over Android platform. The software we developed integrates advantages from several popular IM systems, and provides new prospects for instant message applications.

Keywords

Android Development, Instant Messaging, Mobile Software, Internal Communication

基于Android平台的企业内部 即时通讯应用程序设计

杜新禄, 王星宇

北方工业大学, 北京
Email: duxinlu81@live.com

收稿日期: 2016年11月28日; 录用日期: 2016年12月16日; 发布日期: 2016年12月19日

摘要

企业内员工由于不同的项目分工, 为了提高项目执行效率, 需要合适的沟通和资源互传。本文基于 Android 平台, 通过基本网络通信技术, 实现了一款即时通讯应用程序, 包含了会话、文件传输、资源共享等一系列功能, 为企业内员工沟通提供了良好的工具。本系统的客户端软件对数目众多功能的整合方法为即时通讯类应用的开发提供了新的思路。

关键词

Android 开发, 即时通讯, 手机软件, 企业内部通讯

1. 引言

近年来, 企业项目的构成日益复杂, 项目内各部门之间的即时通讯对项目的执行效率起着重要的作用。其中, 信息的即时性和安全性尤为重要。如今市面上可见的企业内通信应用虽然花样繁多, 但部分存在着功能单一、结构繁冗、需求不对口等缺陷。随着智能手机的快速普及以及移动端软件有着功能强大, 可靠性高, 开发过程中效率高、成本低等一系列优点, 因此本次案例将探讨如何基于 Android 智能平台开发一个企业内部的即时通讯应用软件, 使企业内各部门员工高效而便捷地实现信息交流, 资源共享[1]。

2. 系统设计与实现

2.1. 系统结构设计

由于本系统客户端基于 Android, 因此采用 Eclipse + Android SDK 作为开发, 其中客户端的主要任务是实现用户视图设计与实现, 用户操作的逻辑控制实现, 以及客户端与服务器的网络连接。

系统服务器端、数据库采用 MyEclipse + MySQL 实现, 服务器端的主要任务有: 接收客户端数据请求并转化为服务器编码格式, 为各个数据处理提供程序接口, 从数据库读取数据并相应客户端请求, 使用 Hibernate 技术简化服务器与数据库的数据传输[2]。

本系统应实现的基本功能有:

- 1) 用户在系统中注册、登录。
- 2) 用户可查看消息列表。
- 3) 用户可查看、管理联系人列表。
- 4) 用户可在权限内使用公共社区。
- 5) 用户可编辑并查看自身资料、查看联系人资料。
- 6) 软件功能分布图如图 1, 客户端、服务器实现功能如图 2。

2.2. 系统功能

- 1) 用户注册、登录

用户的账户是系统表示用户身份的唯一标准, 用户通过自身身份才能执行权限内操作、与联系人进行会话, 因此登录是系统不可或缺的功能。用户首次使用本系统需要进行注册, 系统将用户注册信息添加于数据库, 此后用户可登录, 通过系统审核确认身份后, 用户获得账户信息和权限进行操作。

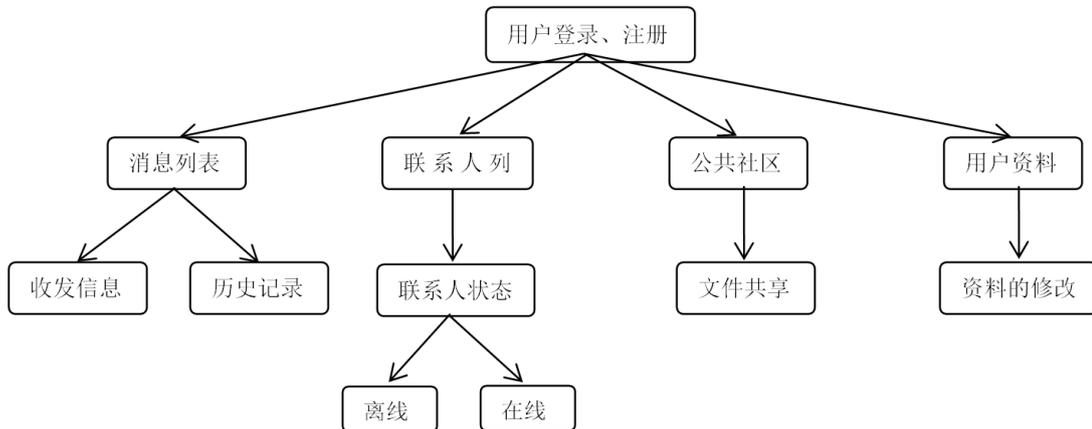


Figure 1. Software function distribution

图 1. 软件功能分布图

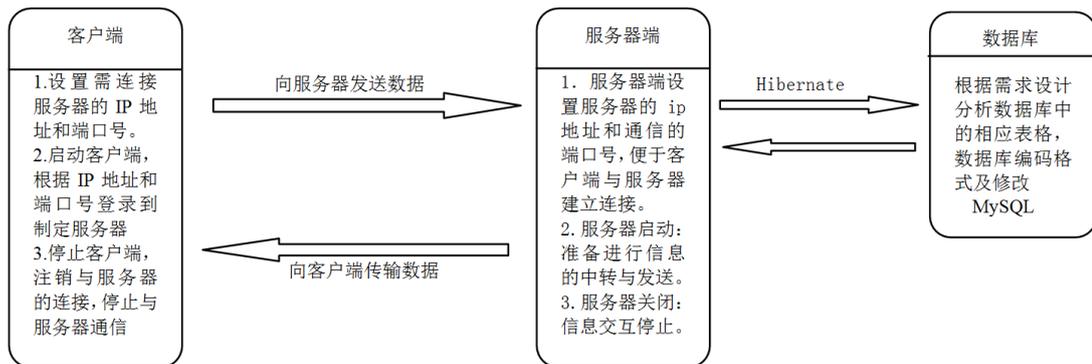


Figure 2. Details of client and server of system

图 2. 系统客户端与服务器端具体功能

2) 消息列表

本系统的主要功能即为信息交流，即时通讯。系统将用户的会话记录存放于客户端，方便用户查看和选择必要的信息。

3) 联系人列表

用户可自主添加项目内相关员工为联系人，添加为联系人之后即可进行会话、文件、图片传输、URL 共享等一系列资源互传的操作。同时，用户可进行查看联系人资料、对联系人进行分组、删除联系人等一系列权限内操作。

4) 公共社区

公共社区是项目内成员的资源共享部分，任何项目内成员都可以查看社区内分享的资料，项目进度等资源，也可以上传项目相关文件，实现了项目内进程即时沟通。

5) 用户资料

用户资料作为展示员工身份的明示，可由员工自主编辑，以便表明身份，联系人之间可以相互查看对方资料。

2.3. 系统模块与实现

2.3.1. 客户端

系统客户端是用户的可视化操作应用层，其视图应美观简洁，功能选项清晰易查，布局结构合理；

其作用应在于完整地收发用户间会话消息, 与服务器进行数据交互, 保存本地资源, 并保证运行时的流畅性和稳定性。

2.3.2. 服务器端

服务器端包含处理程序运算请求的服务器以及保存数据的数据库两部分。其中服务器起到了执行系统逻辑运算的重要作用, 服务器的正常运行是整个系统稳定运行的关键。服务器采用分块分层、保留接口的设计思想, 保证了块与块之间分别执行不同的功能, 互不干扰, 通过接口来实现数据传递, 这种设计减少了代码的重复率, 提高了日后对服务器维护升级的效率。

2.4. 关键技术与算法

2.4.1. 客户的关键技术与算法

客户端(图 3、图 4)根据不同的界面和功能, 对应着各自的运算和代码。客户端采集用户的操作和输入信息, 封装为 JSON 类型数据。JSON 数据相比于 XML 数据的优点是 JSON 数据更易于服务器解析, 层次性更好。通过 TCP/IP 协议与服务器进行网络连接并传输相应数据。客户端采用分块分层的模块化设计思路, 逻辑清晰, 代码针对性强, 易于维护。

具体技术及核心类的构建 UML 如下:

1) 客户端网络连接技术

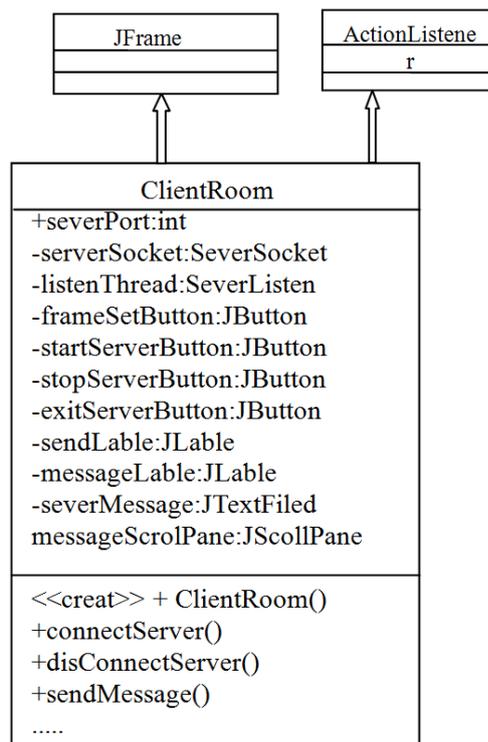


图3注释: ClientRoom继承 JFrame 类和实现 Actionlistener 接口, 封装了客户端界面, 完成客户端与服务器的链接, 断开链接。initClient()方法初始化客户端界面, connectServer()连接服务器 disConnectSever()断开与服务器的链接, sendMessage()发送信息。

Figure 3. The UML of ClientRoom

图 3. ClientRoom 的 UML 结构图

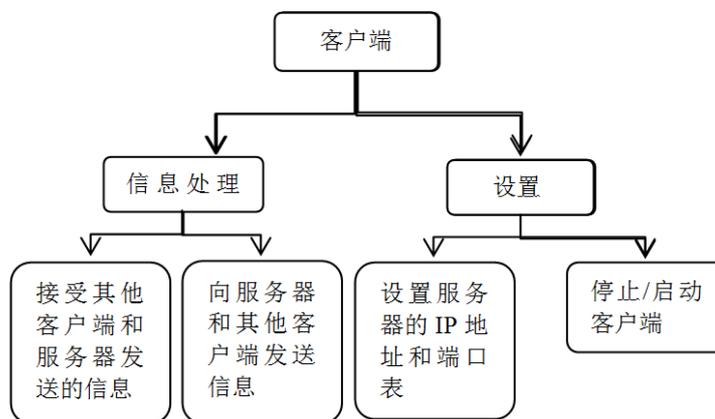


Figure 4. The structure of client side

图 4. 客户端功能结构图

自 Android4.0 系统诞生以后, 考虑到网络安全性, Android 平台禁止开发者在主进程中直接接入互联网。因此在开发过程中, 需要创建新线程来进行与服务器的网络连接, 在该线程中处理数据, 并传入主线程, 保证了主线程的安全性[3]。

线程的创建采用 `java.lang.thread` 包中的 `thread` 类, 与服务器进行连接之后将数据通过创建 `JSONObject` 类的实例进行封装并传输。服务器接收消息后, 解析 `JSON` 对象为可执行、处理的程序和数据。运算后, 服务器以字节流形式将运算结果返回客户端。因为会话中可能包含文字、图片、文件、`URL` 链接等多种信息形式, 因此系统客户端创建了 `ChatMsgEntity` 类来处理不同的消息形式。

2) 会话内容处理

`ChatMsgEntity` 类中主要包含了不同消息形式的接收方法, 其中有文字、图片、语音、文件、`URL` 连接的形式。不同的形式对应着不同的 `Java` 基本数据类型, 通过本类处理后, 再与不同功能的界面相结合, 实现客户端消息的处理。

3) 联系人管理

用户在每次登陆系统时, 服务器端都会发送一个最新的联系人列表至客户端。服务器发送的联系人数据格式统一, 因此只需按顺序进行处理, 得到全部联系人。客户端对联系人进行按照姓名全拼先后的自主排序, 或通过用户的主动分组来产生便于操作的联系人列表。

4) Intent 的使用

系统包含四个不同的层次级功能, 而且要求可以使用户灵活地在不用层次之间快速切换, 因此使用了 `Intent` 来整合了不同的功能界面。`Intent` 对象可以传递给 `Content.startActivity()` 方法或 `Activity.startActivityForResult()` 方法来启动 `Activity` 或让已经存在的 `Activity` 来执行其他任务。`Intent` 对象也可以作为参数将信息返回给调用 `startActivityForResult()` 方法的 `Activity`。通过这种切换处理机制, 保证了用户在不同层次上切换功能的灵活性和使用本系统的流畅性。

2.4.2. 端关键技术与算法

服务器端(图 5、图 6)作为客户端之间发送信息的中转站, 需要接受并转发客户端之间的信息。服务器可以向指定用户发送信息, 也可以向全部用户发送信息。跟踪用户登录状态, 并更新用户数量。

- 1) 采用 `Client/Server` 软件开发架构, 服务器端接受客户发来的信息并转送到对应的客户端。
- 2) 运用 `TCP/IP` 协议的 `Socket` 和 `ServerSocket` 类。
- 3) 在 `ServerHome` 通过 `public ServerHome` 类实现 `ActionListener` 接口, 其中封装了服务器的界面、服

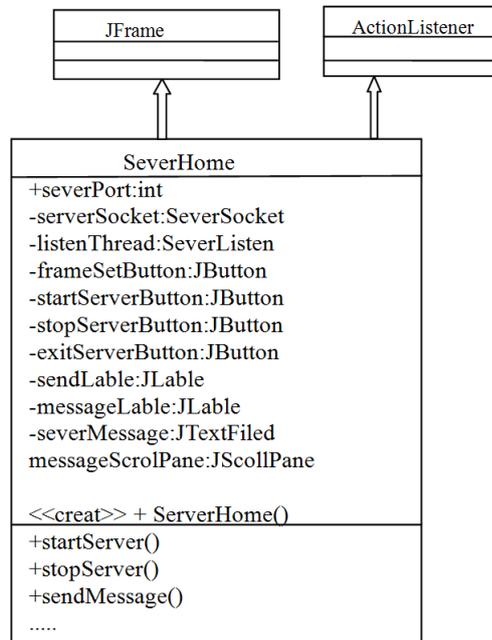


图5 注释: ServerRoom类继承 JFrame 实现 ActionListener 接口, 封装了服务器、服务器端对用户上线与下线的监听。initSever()初始化服务器端的界面, startSever()启动服务, stopServe()停止服务, sendMessage()发送信息。

Figure 5. The UML of ServerRoom

图 5. ServerHome 的 UML 结构

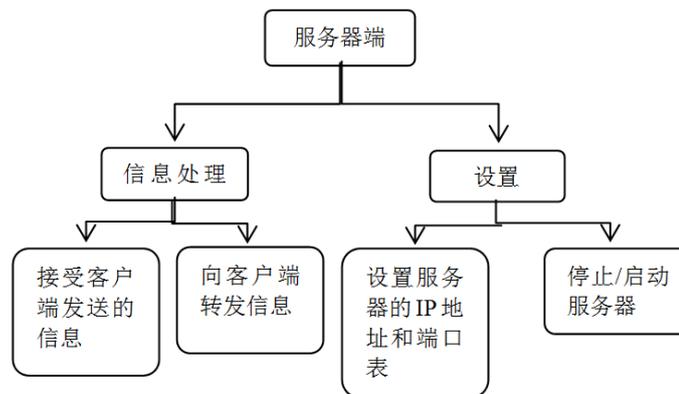


Figure 6. The structure of server side

图 6. 服务器端功能结构图

务器对用户状态的监听、服务器消息的接受发送。

4) 通过 ServerReceive 类来实现服务器的消息收发, 在用户的状态有更新的时候, 比如新用户登陆或用户退出登陆则会更新用户列表, 当用户列表有了更新则会通过 sendUserlist()发送给在线用户和服务器。

5) 通过 SeverListen 类实现 Runnable 接口监听用户的上下线状态。

6) 在 Node 类中封装了用户的的链表节点, 通过链表这种数据结构保存用户信息(用户名、socket 套接字对象等)。

7) 通过 UserInformation 类封装了对于用户信息的一些基本操作, 包括了增删改查的基本操作。

8) 通过 ServerProtConfig 类实现 ActionListen 接口, 包含了初始化, 取消, 保存等操作。

3. 测试

3.1. 测试环境

硬件环境介绍:

1) 服务器端:

处理器: Intel(R)Core(TM)i5-4210M CPU @ 2.60 Hz 2.60 GHz。

已安装内存(RAM): 4.00 GB。

系统类型: 64 位操作系统, 基于 x64 的处理器。

软件部署: 服务器安装了 MyEclipse、TomCat、MySQL。

2) 客户端测试环境:

测试为了模拟真实的软件使用, 即多用户同时使用本系统, 因此将客户端部署在了三台 Android 手机上, 硬件配置如图 7、图 8。

3.2. 测试方案

服务器、客户端均进行合理部署与调试后, 主要观察系统如下运行情况:

- 1) 客户端与服务器之间数据通信是否正常。
- 2) 服务器能否正确解析数据。
- 3) 服务器运算逻辑是否正常。
- 4) 服务器与数据库的连接情况。
- 5) 服务器对多客户端请求的承受能力。

3.3. 测试结果

测试结果如图 9 所示。

	测试机 1	测试机 2	测试机 3
处理器	8 核, 1.8GHz	4 核, 2.15GHz	4 核, 2.35GHz
运行内存	2GB	3GB	6GB
内核版本	3.10.30-00015-g049a08f android@localhost	3.18.20-perf-gfavbe79	3.18.20-perf-g2f72a9d

Figure 7. Basic parameters of test machine

图 7. 测试机的基本参数

	Android 版本
测试机 1	6.0.1MXB48T
测试机 2	6.0.1MXB48T
测试机 3	Android4.4.2

Figure 8. Android version of text machine

图 8. 测试机的安卓版本号

功能	登陆	注册	收发信息	文件共享	查看历史消息
测试机 1	√	√	√	√	√
测试机 2	√	√	√	√	√
测试机 3	√	√	√	√	√

Figure 9. Test result

图 9. 测试结果

3.4. 测试结果及分析

在测试中, 三台 Android 手机上的客户端软件运行流畅, 服务器端“启动服务”, 客户端“登陆”之后, 客户端与服务器端数据传接正常。客户端通过服务器向其他客户端发送数据正常。客户端与服务器连接良好; 服务器端能良好地接收、解析客户端数据, 并准确地进行逻辑运算, 在多客户端请求中表现出了稳定的性能和效率, 我们认为系统很好地符合了预期[4]。

4. 结语

本系统实现了在 Android 平台上的企业内部即时通讯方案, 功能包括会话、资源共享、文件互传, 客户端与服务器的数据传输采用了 JSON 封装的形式; 后台服务器转接信息; 数据库表实现了三范式的设计标准; 在测试过程中, 本系统运行稳定, 性能良好。本系统的实现具有如下良好性质: 功能满足了对即时通讯的需求; 极大地方便了企业内部员工的沟通, 保障了企业内部信息的安全性, 私密性; 层次清晰, 逻辑简明, 易于开发人员的更新维护。

本文详细而准确地介绍了即时通讯类应用程序的开发过程和框架设计, 希望能为同类应用的开发起到抛砖引玉的作用。

基金项目

北京市大学生科学研究与创业行动计划资助项目, 北方工业大学大学生科技活动资助项目。

参考文献 (References)

- [1] 瑞琴, 毕利. 基于 Web Service 的 Android 技术应用研究[J]. 电子技术应用, 2014, 40(1):134-136.
- [2] 王飞, 王创伟. 基于 Android 的数据库初始化方案的研究与应用[J]. 电子技术应用, 2012(10): 55-58.
- [3] 纪晓阳. 线程在 Android 开发中的应用[J]. 软件, 2013(8): 24-26.
- [4] 张立芬, 周悦, 郭振东. Android 移动应用测试[J]. 中国新通信, 2013(3): 84-86.

期刊投稿者将享受如下服务：

1. 投稿前咨询服务 (QQ、微信、邮箱皆可)
2. 为您匹配最合适的期刊
3. 24 小时以内解答您的所有疑问
4. 友好的在线投稿界面
5. 专业的同行评审
6. 知网检索
7. 全网络覆盖式推广您的研究

投稿请点击：<http://www.hanspub.org/Submission.aspx>

期刊邮箱：sea@hanspub.org