

中图分类号：TP3

论文编号：10006XXXX

企业级即时通讯软件的 设计与实施

作者姓名

学科专业

指导教师

培养院系

The Design and Implementation for Enterprise Instant Messaging Software

A Dissertation Submitted for the Degree of Master

Candidate:

Supervisor:

XXXXXX

YYYY University, Beijing, China

中图分类号：TP3

论文编号：

硕 士 学 位 论 文

企业级即时通讯软件的设计与实施

作者姓名				申请学位级别			
指导教师姓名				职 称			
学科专业				研究方向			
学习时间自	年	月	日	起至	年	月	日止
论文提交日期	年	月	日	论文答辩日期	年	月	日
学位授予单位				学位授予日期	年	月	日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写过的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：_____ 日期：_____ 年 _____ 月 _____ 日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名：_____ 日期：_____ 年 _____ 月 _____ 日

指导教师签名：_____ 日期：_____ 年 _____ 月 _____ 日

摘要

企业级即时通讯软件是一种面向企业终端使用者的日常通讯办公工具。与个人即时通信工具相比，是其扩展和延伸，并且在日常沟通的基础上，更加强调集团的组织架构展示、内部文档的信息安全、功能的丰富实用、性能的高可用、消息类型的多样和后台服务接口的扩展。由于集团规模和业务的急速扩展，个人版即时通讯软件已无法满足集团日常办公的需求。同时，集团内部使用的通讯软件 and 市面上流行的几款企业即时通讯产品解决方案，在经过调研后发现无法满足集团未来的办公需求，存在着需求定制化的成本高、周期长，服务器本地化架设维护成本高等问题，与集团发展规划不匹配。在这种情况下，集团决定采用自主研发企业级即时通讯软件并作为产品推广到市场中。

本论文通过论述市场上常用即时通讯软件的优劣及现状，阐述了即时通讯软件终端的设计思想及关键技术。主要包括软件开发流程和质量理论，C/S 基本架构，模块化设计，DirectUI 界面库、插件接口设计、多线程技术等。

在论述理论和技术的同时，本论文还详细分析了即时通讯软件系统的系统需求，系统架构设计、数据库设计、核心界面设计以及核心逻辑模块设计，并详细阐述了终端的测试方案和系统实施的方案和结果。

自主研发的企业级即时通讯软件已于 2017 年通过验收，并成功在集团内部架设和使用。系统已完成对接并导入全部员工信息及组织架构信息，运行稳定，能够满足集团内部的日常办公需求。后序工作中逐渐接入了集团 OA 打卡，OA 会议和邮件等系统，丰富了消息通知的类型。同时，不断收集和总结各个部门的使用反馈，以进行产品后续的升级迭代。

关键字：企业级即时通讯软件、Duilib、模块化

Abstract

Enterprise instant messaging software is a daily communication office tool. Compared with personal instant messaging software, enterprise instant messaging software is an extension of personal version. On the basis of supporting daily communication, he emphasized more on the company's organizational structure input, internal information security, rich and practical functions, high availability of software, diversity of chat content, and extensible background service interface. With the expansion of the company's scale and business, the personal version obviously cannot meet the needs of the daily office. Otherwise, after research, the software already used and several popular product solutions on the market can not meet the company's future office needs, because of the high cost and long cycle of customized demand, high cost of local server installation and maintenance, which does not match the company's future plans. In this case, we decided to adopt the independent research and development of enterprise instant messaging software and promote it to the market as a product.

This thesis, discusses the advantages and disadvantages of instant messaging software in the market, and expounds the design idea and key technology of the software, mainly include software development process and quality management theory, C/S architecture, modular design, DirectUI interface library, plugin interface design, multithreading design, etc.

At the same time of theoretical discussion, this paper also analyzes the system requirements, system architecture design, database design, core interface design and core logic module design of instant messaging software system in detail, and expounds the test scheme of terminal and the scheme and results of system implementation in detail.

Self developed enterprise instant messaging software has passed the acceptance in 2017, and has been successfully installed and used within the company. The system has connected and imported all employee information and organizational structure

information, which is stable and can meet the daily office needs. After that, the system connected to OA clock in, OA meeting and E-mail system, the type of message is expanded. At last, we are collecting and summarizing the feedback of each department to carry out the subsequent upgrade iteration.

Key Words: Enterprise instant messaging software、duilib、modular

目录

第一章 绪论	6
1.1. 课题背景及意义	6
1.2. 国内外研究现状及对比分析	7
1.2.1. 国外研究现状	7
1.2.2. 国内研究现状	7
1.2.3. 对比分析	8
1.3. 研究目标及研究内容	8
1.3.1. 客户端设计	8
1.3.2. 服务器设计	9
1.3.3. 数据库设计	9
1.4. 论文组织结构	9
第二章 系统需求分析	10
2.1. 业务流程分析	10
2.2. 功能需求	11
2.3. 非功能需求	12
第三章 系统设计	13
3.1. 设计原则	13
3.1.1. 禁止在项目中使用的内容	13
3.1.2. 原则	13
3.2. 系统关键技术	14
3.2.1. Duilib 界面库	14
3.2.2. XML 语言	14
3.2.3. Liburl 静态库	15
3.3. 系统架构设计	16
3.4. 系统功能设计	17
3.4.1. 技术架构	18
3.4.2. 通讯协议定义	18
3.4.3. 引擎线程架构	18

3.4.4. 消息处理框架.....	20
3.4.5. 稳定性设计.....	22
3.4.6. 可靠性设计.....	22
3.4.7. 安全性设计.....	23
3.5. 数据库设计	23
3.5.1. 关键数据列表数据库表定义.....	23
3.6. 关键功能设计	27
3.6.1. 基础功能.....	27
3.6.2. 主要功能.....	39
3.6.3. 会话窗口.....	44
3.6.4. 消息管理器.....	48
3.6.5. 设置.....	49
3.6.6. 消息提醒框.....	51
3.6.7. 托盘.....	51
3.6.8. 端通讯录库的加密流程.....	52
3.7. 满足系统性能需求的设计	53
第四章 系统测试	54
4.1. 测试概述.....	54
4.2. 测试目的.....	54
4.3. 测试工具和测试环境.....	54
4.3.1. 服务端测试环境.....	54
4.3.2. 客户端测试环境.....	55
4.4. 测试流程.....	55
4.5. 典型的功能测试用例	55
4.6. 性能测试.....	59
4.7. 测试结果分析.....	60
第五章 系统实施	61
5.1. 实施概述.....	61
5.2. 实施计划.....	61

5.3. 实施组织及培训	62
5.4. 实施环境	62
5.5. 数据准备	64
5.6. 系统上线	64
5.7. 实施效果评估	65
第六章 总结与展望	66
参考文献	67
致谢	68

图目录

图 1 业务流程图.....	11
图 2 架构设计图.....	16
图 3 功能模块划分图.....	17
图 4 消息引擎时序图.....	20
图 5 主界面.....	28
图 6 会话界面.....	29
图 7 消息管理器.....	29
图 8 安装流程图.....	30
图 9 升级流程图.....	31
图 10 登陆、退出流程图.....	32
图 11 消息订阅流程图.....	33
图 12 单人会话流程图.....	33
图 13 多人会话流程图.....	34
图 14 固定群会话流程图.....	34
图 15 会话成员变更流程图.....	34
图 16 订阅流程图.....	35
图 17 组织架构更新流程图.....	35
图 18 自定义组更新流程图.....	35
图 19 最近会话更新流程图.....	36
图 20 会话定时更新流程图.....	36
图 21 在线状态更新流程图.....	37
图 22 自动重连流程图.....	39
图 23 登录状态变更流程图.....	40
图 24 更新消息流程图.....	41
图 25 组织架构更新流程图.....	42
图 26 消息发送流程图.....	43
图 27 会话管理流程图.....	45
图 28 文件发送流程图.....	46

图 29 更新回执流程图.....	48
图 30 个人设置更新流程图.....	50
图 31 数据库加密流程图.....	52

表目录

表 1 基础接口定义类表.....	18
表 2 插件接口类图.....	20
表 3 事件管理器接口类图.....	21
表 4 登录信息表.....	23
表 5 最近更新时间表.....	24
表 6 部门表.....	24
表 7 员工信息表.....	25
表 8 常用联系人.....	26
表 9 常用部门.....	26
表 10 会话表.....	26
表 11 日程信息表.....	27
表 12 服务器测试环境.....	54
表 13 客户端测试环境.....	55
表 14 登陆界面功能测试用例.....	56
表 15 单人聊天功能测试用例.....	58
表 16 性能指标.....	59
表 17 测试分析报告.....	60
表 18 实施工作计划.....	61
表 19 实施人员准备.....	62
表 20 硬件环境配置表.....	63
表 21 软件环境配置表.....	错误!未定义书签。

第一章 绪论

1.1. 课题背景及意义

互联网时代充满了机遇，特别是在“互联网+”的推动下，集团内部的业务发展蒸蒸日上。随着集团的规模扩大和并购扩张，部门之间和同事之间的信息传递与沟通方式直接影响了日常的办公效率。经过调查发现，有的部门使用个人即时通讯软件进行沟通，比如QQ和MSN等；有的部门规模较大则引入了第三方即时通讯软件，比如企业QQ；在这样的背景下，不仅部门之间的沟通成本高、后台信息数据不共享，且存在极大的信息安全隐患，一旦造成机密信息泄漏，会给公司造成损失。另外，集团内部存在多个其他办公系统，比如OA系统、人力资源管理系统、E-Mail系统等，需要在不同的系统之间建立联系并实现自动化的消息通知。综上所述，使用一款符合集团特点的企业级即时通讯软件系统的需求被提出。

企业级即时通讯软件是个人版本的扩展和延伸，在支持日常沟通的基础上，减少了娱乐功能，更强调集团的组织架构展示、内部文档信息的安全、功能的丰富实用、性能的高可用、消息类型的多样、后台服务接口的可扩展（包括但不限于日常OA打卡、企业公告推送、会议信息推送、企业组织架构、通信回执、邮件通知等功能）。同时，集团内部使用的通讯软件与当前市面上流行的几款企业即时通讯产品解决方案在经过调研后无法满足公司未来的办公需求，原因主要包括以下几点：

1. 购买和使用第三方软件会导致每年产生巨额的定制费用、服务器维护费用，并且存在一定的工程周期，若未来集团架设了新的系统需再次定制，从发展角度考虑此方案的性价比不高，并且不能适应集团未来发展的规划。
2. 个人通信软件无法满足企业信息安全管理要求，如QQ、WeChat、Skype等。因为聊天数据服务器并未架设在集团内部，缺少了消息审核和记录，极易造成企业泄密并无法追溯源头等问题；另外，个人版本的聊天好友仅支持搜索添加，对于不了解集团组织架构的同事很难即时找到沟通目标，并且缺少个人办公信息的展示（如办公座机、办公邮箱等），间接降低了办公的效率。
3. 集团内部的日常办公软件不止即时通讯软件一种，迫切需要一个平台将各个系统打通，实现单入多出的访问逻辑。而即时通讯软件作为日常使用量最高的软件非常适合这个核心角色。所以，自主开发并链接后台接口不仅能够节约定制

成本，更有利于集团内部信息和系统访问权限的安全。

4. 提供后台扩展接口，支持接入并广播自定义消息，比如日常新闻、会议提醒等推送消息。
5. 自主研发可以设计符合企业文化形象的界面和皮肤，体现集团文化。
6. 目前集团内部存在一些部门使用了第三方即时通讯软件，造成了消息数据无法共享，组织架构分散，用户信息不统一等问题。需要将已有后台服务器之间相互打通，实现跨地区跨部门的通信。
7. 集团计划丰富 IT 产品线，计划将本产品推向市场并盈利，根据最终实施效果进行评估。

在这种背景下，集团决定采用自主研发企业级即时通讯软件，并期待以此为契机打造一款产品最终推广到市场中。我作为 PC-Windows 开发组成员全程参与了项目建设工作。

1.2. 国内外研究现状及对比分析

1.2.1. 国外研究现状

大名鼎鼎的美国社交网站 Facebook，曾经在 2013 年也发布过基于 Windows 桌面端的聊天软件“Facebook Messenger”，也就是从此开始了对即时通讯软件的布局。并在软件发布的次年 2 月收购了跨平台移动消息公司 WhatsApp 并通过了欧盟反垄断监管机构的批准，总价值达到了约 190 亿美元，足以体现对即时通讯领域的重视。WhatsApp 的全名为 WhatsApp Messenger，该软件基本支持了市场上所有流行的移动系统，Facebook 也借此打开了移动端的即时通讯市场。借助于 Facebook 强大的社交技术和资源，WhatsApp 站稳了市场，同时已经支持发送和接收信息、图片、音频文件和视频信息等多种类型的消息，丰富了使用场景。

经过调研，WhatsApp 首先是 UI 上不符合国内用户的习惯，即使设计上确有优势但却间接提高了用户学习成本；其次该软件的核心还是主打社交和娱乐，缺少对企业日常办公中环境的支持；最后，该软件以移动端为主，缺少基于桌面端的布局，不符合工作需求。

1.2.2. 国内研究现状

目前，国内知名的企业级即时通讯软件有很多，其中使用最多系统应该就是腾讯企业版 QQ 软件。

基于已有的个人版功能，腾讯公司为中小企业搭建企业级即时通讯工具推出了企业

版 QQ 软件。主要立足于企业管理，简化内外部沟通的难度，提高日常办公效率。其理念是“安全、高效、可管理”，打通了与个人版 QQ 软件用户之间的联系。

其次是最近流行起来的钉钉，是阿里巴巴集团专为中国企业打造的免费沟通和协同的多端平台，提供 PC 版，Web 版和手机版，帮助中国企业通过系统化的解决方案（微应用），全方位提升中国企业沟通和协同效率。钉钉以内部通讯为基础，增加了报表、考勤、邮件通知、短信提醒等日常办公的功能，可以说基本满足了员工对办公的需求。最新的“DING 一下”功能是一大亮点，实现了重要消息催促和置顶的功能。

综上所述，结合当前公司的需求考虑，以上解决方案并不能完全满足大型企业管理，且仅限于聊天等沟通手段，缺乏其他符合公司特点的日常办公的能力，比如预留公司内部其他系统对接接口、会议提醒等日常推送信息。对于以上解决方案中提供的定制化的服务及其对应产生的维护时间成本，均不符合公司的发展战略。

1.2.3. 对比分析

目前，虽然国内外主流的个人版和企业版即时通讯软件基本都可以满足日常通讯需求，但是并不能满足本集团的特点和定制化需求，缺少一些必要的功能或者多出一些无用功能的收费。

为了避免以上问题并控制成本，决定吸取现在市场中主流企业版本的即时通讯软件产品的特点和功能，独立设计一款适用于自己公司的聊天软件并打造为一款产品，要从实际功能上符合企业内部的办公需求，并为后续推广做准备。

1.3. 研究目标及研究内容

1.3.1. 客户端设计

客户端界面开发采用基于 DirectUI 库的 Duilib 界面库，Duilib 可以将用户界面和处理逻辑彻底分离，极大地提高用户界面的开发效率。其特点是所见即所得，基于 GDI 的自绘彻底解决了使用传统 MFC 界面库开发软件不美观、界面细节处理不好、使用硬编码、开发效率低下、生成程序体积大等问题。

界面控件布局使用 XML 语言描述界面风格，XML 语法简单，易于学习和掌握 支持改变皮肤色调，支持大多数控件透明，支持 png、bitmap、 gif、 ico 等图片格式。

设计方案参考 QQ 个人版，分别设计登陆界面、主菜单界面、多 Tab 聊天窗口界面以及历史消息记录界面四部分。

1. 登录界面以简洁直观为主，需要提供账号和密码的输入窗口，并支持保存密码和自动登陆功能，登陆按钮置于界面底部中间。登陆界面的设计中还应包含升级提示窗口，负责升级包的下载和自动安装并提供图形化界面显示。
2. 主界面采用垂直布局，其中最上方设计用于显示个人照片和信息、H5 微应用按钮以及其他办公常用快捷按钮；下方采用树形结构展示聊天对象及分组，包括最近联系人、常用联系人、常用群组和组织架构树形结构。
3. 聊天窗口界面采用左中右分栏模式：左边显示聊天列表 TAB 标签结构；中间采用垂直布局用于展示聊天内容，包括消息发送窗口和常用功能菜单栏（包括文件传输、截图、字体设置、消息回执、查看历史消息）；右边显示头像、个人（或群组）信息的列表。
4. 消息记录窗口采用左右分栏结构设计，左边显示所有对话记录列表，右边采用垂直布局，展示消息过滤等功能按钮和对应历史聊天内容窗口。

我主要负责客户端所有界面的设计和对应逻辑层实现，并参与后台和服务器通信接口的协议开发工作。系统在架构设计上按照功能划分模块，所有模块都以接口方式提供功能，将每一个模块称为一个组件。

1.3.2. 服务器设计

服务器的设计需要实现基于 Socket 接口通信、消息中心、数据中心等功能，同时需要负责接入集团内部系统，如 OA 系统、E-Mail 系统、会议提醒等。

我参与的服务器设计主要是以协议开发为主，该协议制定了一套二进制的收发逻辑。协议内容包括登陆模块、聊天模块、组织架构同步模块等

1.3.3. 数据库设计

数据库设计原则以简单高效为主，使用简洁高效的 SQLite 数据库，便于提高查询效率和维护。

我负责并参与数据库中聊天数据表结构设计，使用已有的第三方接口库实现数据的写入和读取。

1.4. 论文组织结构

第一章主要讲述了集团中对即时通讯软件的需求，并通过分析和对比当前流行的企业级即时通讯软件，阐述了项目对集团发展的意义。

第二章详细讲述了集团内部的需求，以及对功能架构进行的设计方案。

第三章详细讲述了每个功能模块的设计和实现方案。

第四章详细讲述了测试方案和部分测试用例。

第五章详细讲述了项目产出的系统的部署方案。

第六章是对整个项目经历和结果的总结，以及对该项目后续开发的展望。

第二章 系统需求分析

2.1. 业务流程分析

企业级即时通讯软件，首先应保证每个员工 ID 及对应的账号信息唯一即为 UUID，账号模块是整个软件的基础，用户登陆后会被服务端分发唯一的 Token ID，用于验证后续通讯时的身份信息。

员工在登录后会基于历史聊天记录自动订阅和更新聊天信息，登陆后可以通过组织架构选择对应聊天对象进行聊天消息的收发操作。除基本功能外，还支持组织架构搜索、历史聊天消息查询和待办流程消息的订阅推送，具体业务流程详见图 1。

直到用户退出软件，整个业务流程结束。

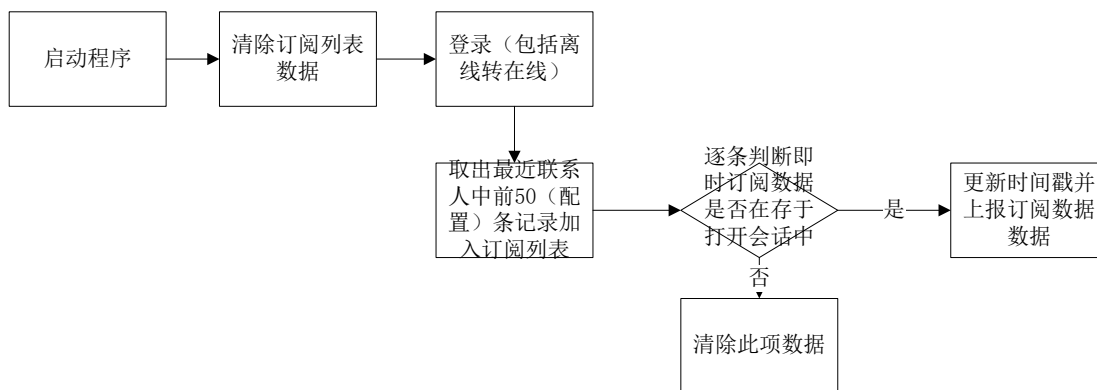


图 1 业务流程图

2.2.功能需求

功能需求是软件的核心需求，以下简单介绍几点核心功能需求：

1. 账号管理：

每个用户分配独立账号和 ID，称为 UUID，用于集团内部部署的服务器进行登陆和认证，同时用于验证后续产生的每条聊天信息的对象。

每个账号中都应保存该用户的基本信息，应包括中文姓名、英文姓名、性别、姓名拼音、出生日期、职务、电子邮箱、办公电话、移动电话、个性签名、地址等。其中允许用户在个人设置页面上传头像，如无头像显示默认头像。

2. 即时通讯：

即时通讯是本软件的基础功能，也是核心功能，支持单人聊天、多人聊天和群组聊天，聊天内容应包括文本聊天、图片传输、文件传输和表情等，用户聊天信息在投递到服务器之前会进行加密，保障安全。支持历史聊天记录查阅功能。应聊天消息类型支持扩展，为后续自定义消息推送等扩展功能预留接口。

3. 组织架构：

组织架构类似于好友系统，但企业级即时通讯软件应自动导入集团的组织架构并以树形图的形式在界面中，用户可以通过查询或者手动选择的方式查找聊天对象，并且服务器后端应支持人力资源管理系统组织架构录入。

4. 办公消息推送：

推送形式采用屏幕右下角弹窗形式展示给用户，提供广播消息、待办消息等与办公相关的公告消息的推送。

5. 快速访问按钮:

在界面中提供特定区域展示公司内部系统的快捷访问按钮, 支持快速打开对应集团系统的主页, 同时完成自动登陆。该按钮组合方便员工随时随地快捷访问公司内部其他系统。

2.3. 非功能需求

非功能性需求主要是针对软件运行期间的高可用需求, 以保证系统的稳定可靠, 减少崩溃等异常问题。主要包括以下几点:

1. 安全性:

数据安全和信息安全自主研发的核心需求, 聊天信息在网络通信中需要进行加密, 保证企业内部信息不外泄。服务端需定时为进行业务数据和用户数据备份, 对于可能出现的错误和故障, 建立异常捕获和快速响应机制, 减少因不可抗力导致信息财产损失。

2. 可扩展性:

服务端需预留接口, 用于后续接入第三方功能, 如视频通话服务模块。同时客户端软件在数据结构的设计上要进行良好的抽象设计, 严格把控类的继承, 主要用于支持自定义聊天信息类型及其他消息通知类型的扩展。

3. 兼容性:

为了适应集团当前硬件环境, 在 PC 端软件的开发和迭代过程中需要保证向下兼容, 需要支持 windows XP、windows 7、windows 8、windows 10、winServer 2012、winServer2016 等办公常用系统。对于 Mac 和移动端的支持, 本论文暂不涉及。

4. 用户体验:

保证 UI 的交互设计符合多数用户习惯, 降低学习成本, 同时主界面应提供快捷按钮用于快速链接至企业内部其他系统平台。

会话窗口的工具栏支持常用功能, 如字体设置、截屏操作、文件传输等。

UI 设计色调和 Logo 图片能够体现企业文化。

第三章 系统设计

3.1. 设计原则

软件开发中涉及到诸多流程和环节，应遵循软件质量管理基本理论。同时为了方便团队成员之间的相互协作和代码 Review，基于代码安全性、可读性及可扩展性，制定了编码规范，由团队每个成员遵守。另外，在开发过程中需要同步建立良好的结构和完备的文档，使功能易于理解、系统易于扩展。

在满足项目质量管理的基础上，需根据业务逻辑进行模块划分，降低模块之间的耦合，对项目后续持续迭代有积极影响，保证项目的健壮性、稳定性。

项目开发使用 SVN 进行版本迭代工具。代码提交需要指定一名成员进行 Review，在通过后方可 Merge 到主分支。

3.1.1. 禁止在项目中使用的内容

1. 禁止使用当前路径(.)及与路径相关 API。
2. 禁止在模块之间的参数传递对象，只能使用指针或者引用的方式。
3. 禁止在复合体中包含对象成员，在模块之间传递。
4. 禁止代码批量迁移、复制。
5. 禁止使用 DLL 隐式链接本地库。
6. 禁止使用自定义全局变量、硬编码、魔法数、拼音与拼音缩写。
7. 禁止在一个类中直接访问另一个类的成员变量。
8. 禁止使用 goto 语句。
9. 禁止不同版本模块混合调用。
10. 禁止多级指针连续引用。

3.1.2. 原则

1. DLL 模块均以接口提供的形式呈现，由 API 导出接口。
2. 符号命名使用匈牙利命名法，从变量的名称可以看出：作用域、类型、功能。
3. 良好的注释习惯，添加、删除、修改等等相关注释。
4. 每修复一个 Bug 进行一次提交

5. 提交时要填写相关的说明内容，且保证可以通过编译。
6. 第三方库的隐式链接需要指定库名，如：sqlite3.lib。
7. 静态链接均使用指令进行链接，如：#pragma comment(lib, “zip.lib”)。
8. 运行时库使用静态库版本。

3.2. 系统关键技术

系统的服务端和客户端均使用 C/C++ 语言作为后台逻辑实现的开发语言。

3.2.1. Duilib 界面库

Duilib 界面库是基于 Viksoe 的 DirectUI 项目，并且以 BSD 协议开源。DirectUI 项目的特点，是所有子窗口都是逻辑上的窗口，直接使用 GDI 绘制于父窗口之上。所以任何类似于 SPY++ 的桌面窗口捕获工具都无法定位到子窗口。Duilib 库在原有项目基础上进行了二次开发，设计了很多易于我们使用的接口。

基于界面层和逻辑层分离的核心思想，引入了 XML 语言来描述控件的布局。每一个界面都可以被一个 XML 文本描述。同时还提供自动化设计布局的工具软件辅助，显著降低了界面和皮肤的开发成本。

其次，GDI 自绘的思想除了在性能上存在极大优势以外还有安全性上的优势。对于使用 RichEdit2.0 库建立的聊天窗口，无法被任何 HOOK 进程均捕获句柄，无法定位关键 API 则无法捕获到内存中的聊天数据，有效的提高了软件的安全性。

最后，Duilib 界面库支持嵌套 Web 页面，兼容 ActiveX 控件，处理 Flash 的问题得到了解决。同时可以和 MFC 库等其他界面库相互混合使用，降低了界面库的升级和切换的成本，当后续有对界面库进行替换的需求时，也会变得容易开发。

3.2.2. XML 语言

XML 是一种用于标记电子文件使其具有结构性的标记语言。标记是指程序所能理解的符号，用于处理计算机之间的各种信息。标记的类型没有具体限制，可以用来标记数据或定义数据类型，是一种允许用户自定义的源语言。Duilib 库支持自动对 XML 文件各个节点的解析，并通过控件 ID 建立与 C++ 代码的联系，同时自动进行 GDI 界面的绘制。

XML 是类似于 HTML 的纯文本格式语言，其每个节点元素包含开始标记、结束标

记和内容。在本项目中，XML 用于描述控件在画布中的控件 ID、坐标、长度、宽度、皮肤、字体、消息文本内容等。

Duilib 库处理 XML 解析和编译时，会在 Debug 版本中触发断言报错，Release 版本中会放弃绘制该行控件。

XML 语言有如下基础规则：

1. XML 声明语句 XML 声明是文档的第一句，其格式为 “<?xml version="1.0" encoding="utf-8"?>”。
2. 在 XML 文档中区分大小写，需要保持一致。
3. 每一个 XML 文件中具有唯一的根元素，用于包含其余子元素，需要在声明之后创建，并在所有子元素结束标记之后创建根元素结束标记
4. XML 的节点元素的属性值需加引号（可以是单引号，也可以是双引号），否则该属性将被忽略并视为异常。
5. XML 的标记必须成对出现，即所有的标记必须有相应的结束标记。

3.2.3.Libcurl 静态库

Libcurl 静态库是封装了 curl 命令和函数的一个跨平台库（curl 是一个利用 URL 语法在命令行下工作的文件传输工具），支持的通信协议有 FTP、FTPS、HTTP、HTTPS、TFTP、SFTP、Gopher、SCP、Telnet、DICT、FILE、LDAP、LDAPS、IMAP、POP3、SMTP 和 RTSP。

Libcurl 静态库的使用上需要用户传入回调函数地址或者闭包任务完成数据交互，在每一个任务结束时，Libcurl 将调用回调函数。使用时需注意使用 curl_global_init()初始化和 curl_easy_cleanup()结束任务并释放内存，防止内存泄漏。

在本项目中，已完成对 libcurl 静态库的二次封装并实现了接口，网络数据传输操作均调用该接口实现基于 HTTPS 协议的 socket 通信。

3.3. 系统架构设计

根据项目需求，将系统模块自上而下划分为三层，分别为：

UI 层：负责页面的加载展示和界面操作的实现，每一个页面的实现分别由对应的插件实现，不负责业务逻辑的处理，通过消息引擎与业务逻辑层进行交互，该模块简称 **EIM**。

业务逻辑层：负责处理核心数据和业务逻辑，是软件的核心引擎，负责所有消息事件的分发，包括页面事件和底层事件，主要包含主引擎 **EIMEngine**；

核心功能层：分别为各个主要功能的 **DLL** 模块，运行时会作为插件被加载。主要包含账户信息 **Agent**、数据库、聊天数据信息等。

详情如图 2 所示。

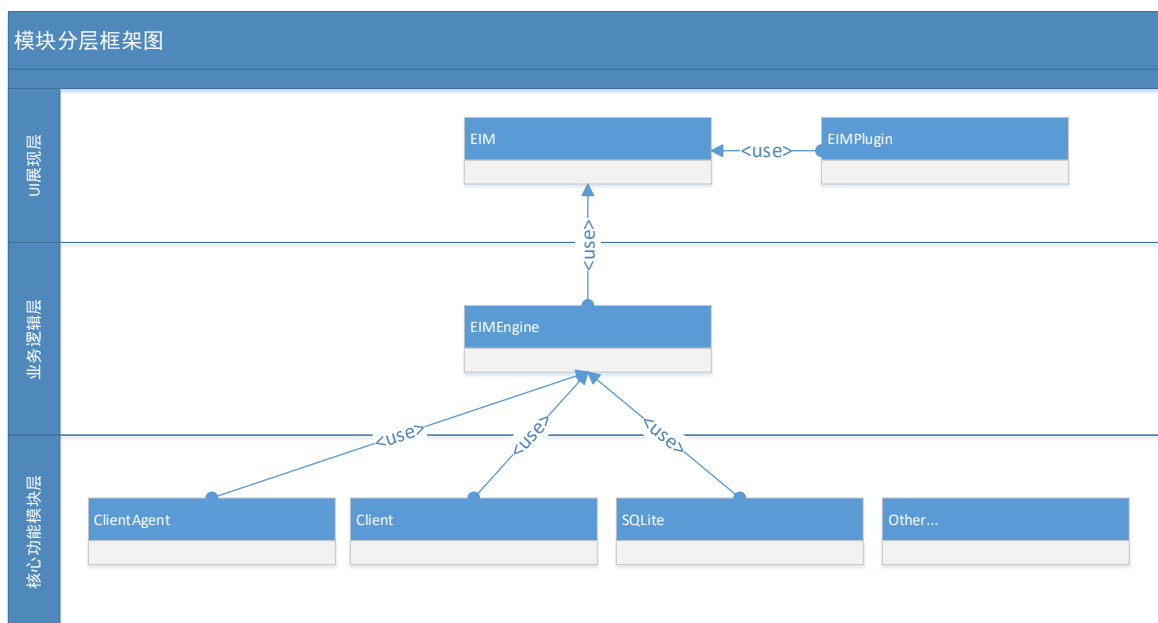


图 2 架构设计图

3.4. 系统功能设计

基于架构设计，将三个层次进行二次分解，即每一个模块需要实现的具体功能如图 3 所示。

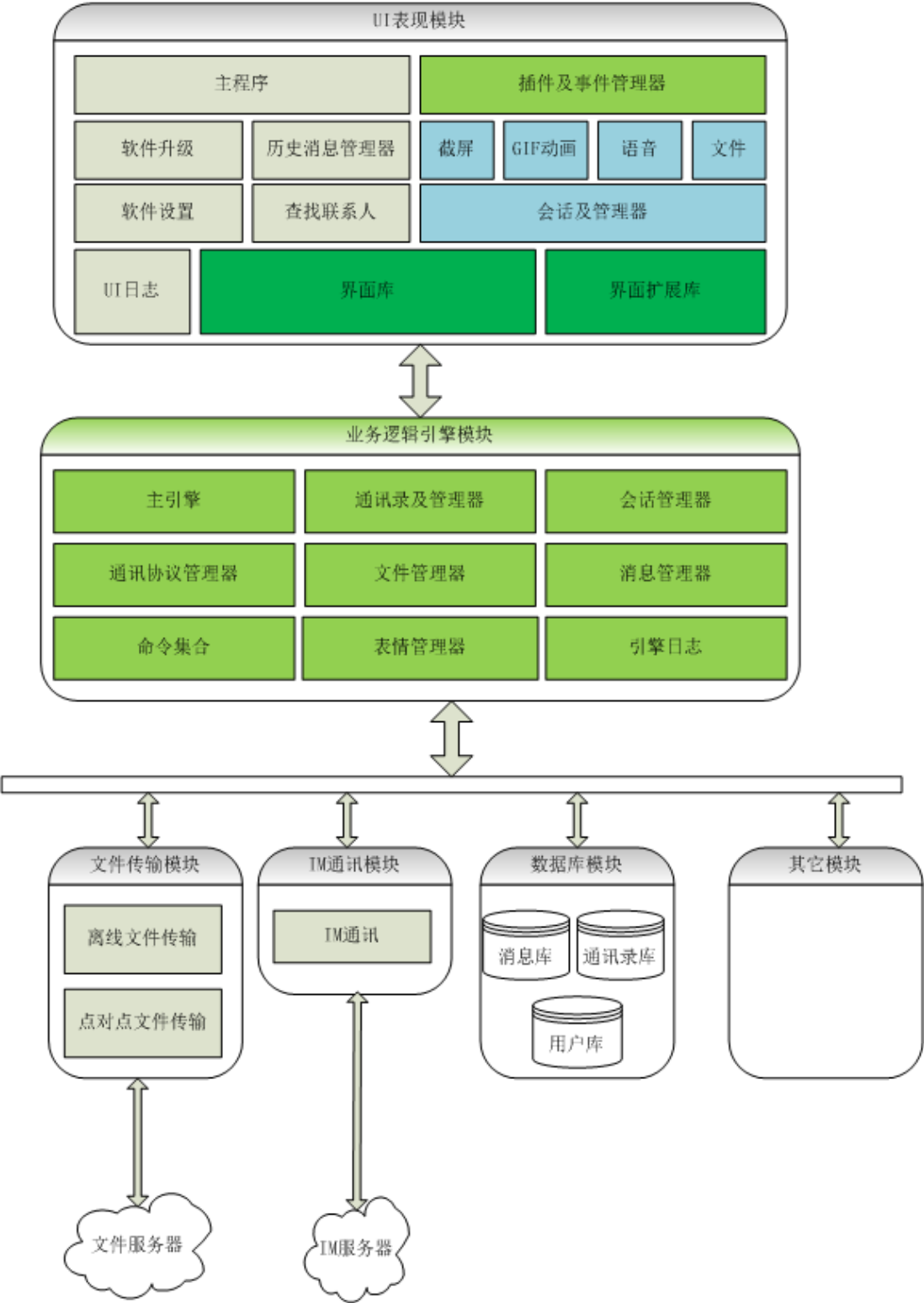


图 3 功能模块划分图

3.4.1. 技术架构

技术架构是设计的前提，一个规范的优秀的技术架构可以提高软件性能并降低开发难度。本项目中技术的设计和使用，主要基于以下三点：

1. 基于面向对象思想，每个组件均是一个接口，为所有的类定义一个虚基类。
2. 接口实现引用计数，避免内存泄漏。
3. 接口的实例化遵循创建→查询→函数调用→对象释放的步骤。

该设计方案便于不同接口之间的相互调用，同时可以防止内存泄漏的发生。基础接口类设计为虚基类，其定义参见表 1。项目中任何模块的任何对象的定义均需继承此虚基类。

基本接口定义: I_EIMUnknown	说明
HRESULT QueryInterface(const TCHAR* pclsIID, void** ppvIObject);	创建子接口
ULONG AddRef(void);	引用计数加一
ULONG Release(void);	引用计数减一、释放对象
普通接口对象只需要实现 Release，以正确释放资源；	

表 1 基础接口定义类表

3.4.2. 通讯协议定义

项目的通讯协议基于 HTTPS，主要用于与服务器进行数据交互。为了降低网络负载和字符串解析性能问题，需依据以下四点：

1. 网络传输的数据使用 XML 格式
2. 界面 UI 的消息传输格式使用 XML 格式，且异步发送到引擎进行处理；
3. 任何模块的工作线程只支持命令接口。任何功能项的实现均对应一个独立的命令；
4. 所有命令之间没有任何逻辑上的关联，相互独立。只有协议上的关联，如有发送消息命令：CmdSendMsg，它有对应的应答 CmdSendMsgAck。

3.4.3. 引擎线程架构

引擎根据功能划分为 5 个独立的线程用于处理消息。每个引擎的工作线程相互独立

又相互关联，通过消息传递的形式实现事件的通知，实现流程参考时序图图 4。

每个引擎的具体实现功能如下。

1. **MgrThread**: 管理者线程，监视管理者线程，再次唤起已经退出的工作者线程；
2. **SendThread**: 命令发送线程，专职负责命令的发送。如果有需要数据保存，再将命令入队数据库队列；否则如果通知 UI，则将命令入队 UI 队列；
3. **RecvThread**: 命令接收线程，专职负责命令的接收。如果有需要数据保存，再将命令入队数据库队列；否则如果通知 UI，则将命令入队 UI 队列；
4. **DbThread**: 数据库线程，专职进行数据的写入。如果通知 UI，则将命令入队 UI 队列；
5. **DispThread**: UI 分发线程，专职将命令结果通知 UI。如果有需要再次发送命令（主要在文件、图片发送时），再将命令入队命令发送队列；否则，如果要入库，则将命令入队数据库队列。

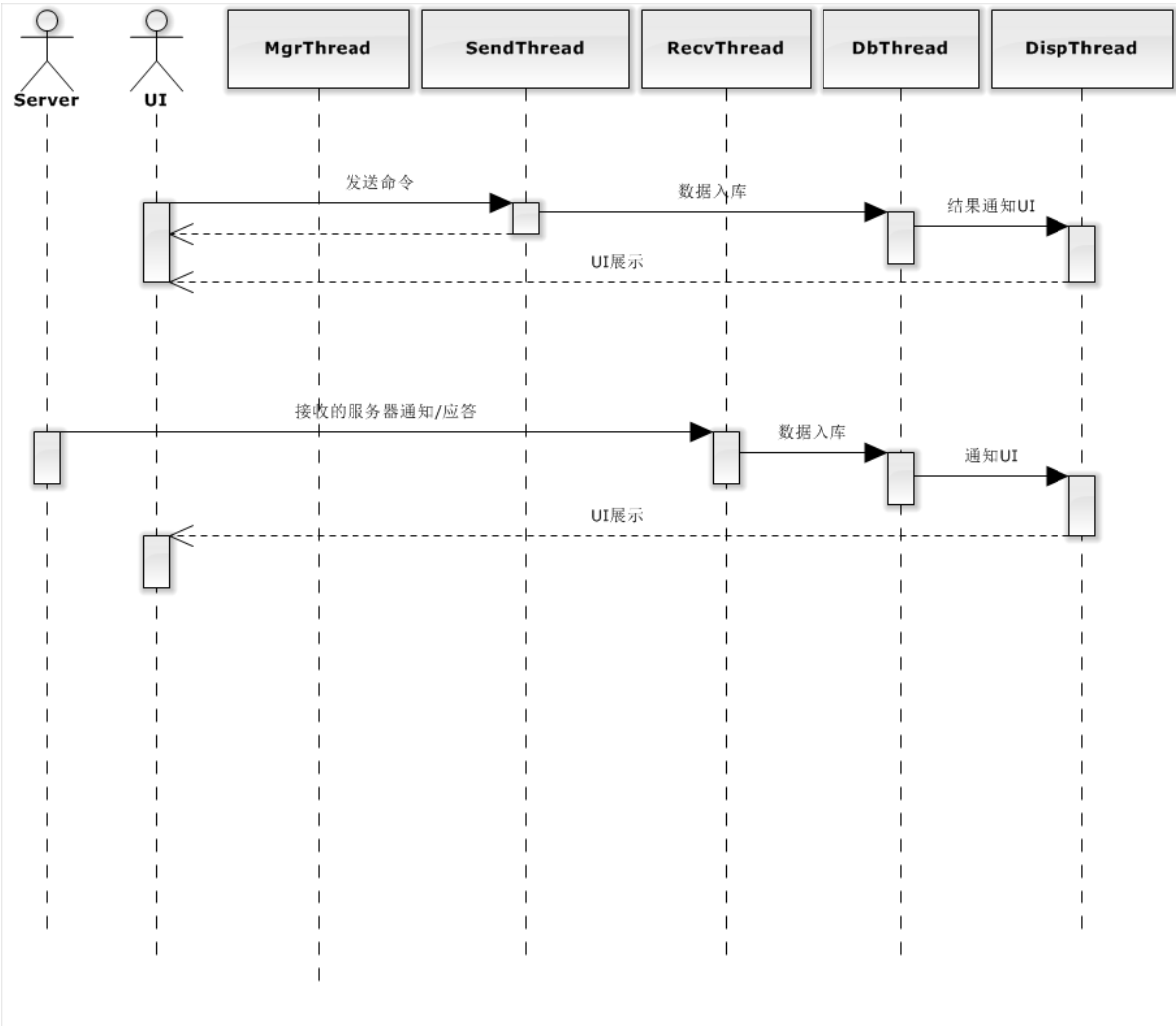


图 4 消息引擎时序图

3.4.4. 消息处理框架

所有功能库，由插件管理器来进行消息的管理。任何插件，必须在 `InitPlugin` 或之后进行注册事件，在 `FreePlugin` 或之前反注册事件。事件的注册与反注册绝对禁止在类的构造函数与析构函数中进行。

插件管理器的接口定义参照表 2。

表 2 插件接口类图

插件接口定义: I_EIMPlugin	说明
<code>BOOL InitPlugin(I_EIMEventMgr* pEventMgr, LPCTSTR lpszConfigFile);</code>	初始化插件
<code>void OnEvent(...)</code>	插件事件响应

<code>unsignedint u32Event, void* lpvParam)=0;</code>	
<code>BOOL GetPluginInfo(PS_PluginDllInfo pPluginInfo)=0;</code>	获取插件信息, 插件信息在注册的配置文件中优先显示。
<code>void FreePlugin()=0;</code>	释放插件
接口定义兼容继承于的 I_EIMUnknown, 所以要实现 I_EIMUnknown 的相关函数;	

事件管理接口会在插件管理器的功能初始化 `InitPlugin` 参数中传入, 有需要事件功能的就保存接口指针 `I_EIMEventManager`。事件的发送者不关心谁接收, 有几个人接收; 接收者也不用关心是谁发送, 如何发送;

接口定义参照表 3。

表 3 事件管理器接口类图

插件接口定义: I_EIMEventManager	说明
<code>virtual BOOL RegisterEvent(DWORD dwEventID, I_EIMPlugin* pPlugin,</code>	注册事件, 事件范围: [0x000 ~ 0x100) 保留为 ClientAgent 使用 [0x100 ~ 0x200) 保留为引擎内部使用 EVENT_APP (0x00000200): 为 eIM 内部使用开始范围

E_EIMEventType eType)=0;	EVENT_USERS (0x00008000)：为第三方插件开发使用开始范围 必须在 InitPlugin 或之后进行注册事件（即可以在需要时再注册）
virtual BOOL UnregisterEvent(DWORD dwEventID, I_EIMPlugin*pPlugin)=0;	反注册事件 必须在 FreePlugin 或之前反注册事件（即可以在不再需要时反注册）
Virtual void SendEvent(DWORD dwEventID, Void *ppvParam)=0;	发送事件，事件参数类型由发送方定义，事件接收方须按照发送方的下定义响应与使用事件参数
接口定义兼容继承于的 I_EIMUnknown，所以要实现 I_EIMUnknown 的相关函数；	

3.4.5. 稳定性设计

软件的稳定性是设计中的主要关注点，稳定性直接影响使用者的体验，本项目中需要考虑异常重启、DMP 文件收集、数据库多线程读写的功能

系统中定义由守护线程，负责随时监视、管理工作线程。当任何工作线程发生异常时由守护线程进行重启操作。当系统发生崩溃时，目前还没有通过注册 Windows 服务的形式进行重启，后续会逐渐实现。

DMP 文件的收集通过系统 API 函数 SetUnhandledExceptionFilter 实现，自定义异常捕获函数中会收集 Minidmp 类型的信息以。根据对应版本号和对应 pdb 文件定位异常原因。后续会实现 DMP 日志的自动上传和筛选功能。

SQLite 数据库不支持多线程同时写，所有的入库操作，都必须在数据库 DB 线程中进行，由多个线程同时读写数据库事件的存储队列，减少锁的使用。

3.4.6. 可靠性设计

软件和可靠性与健壮性，第一要从编码的规范进行要求；第二要从架构的组织进行管理。编程人员需要遵守编码规范，依照架构设计进行接口的实现，防止内存泄漏等异常问题。同时项目采取模块化的设计思想，保证模块间的独立性，可以有效增加软件的可靠性，并对问题的定位和分析有着积极的意义。

3.4.7. 安全性设计

1. 数据库中将用户的密码进行加密保存，且数据库本身会进行加密。
2. 数据通讯之前，消息体本身进行对称加密。
3. 使用 RichEdit2.0 实现聊天窗口，避免基于句柄的窗口消息获取。

3.5. 数据库设计

系统使用 SQLite 数据库作为数据的存储库，为保证数据完整性和可读性，对于存储数据的格式做以下约定：

1. 键、索引字段均以下划线结尾，不再作特别说明；
2. SQLite 是弱类型，字段变长，类型的定义主要用于描述协议。
3. flag 字段为位标识，用于功能扩展标识。
4. timestamp 字段为时间戳
5. orders 字段为用于排序，服务器给的 0 为无序

3.5.1. 关键数据列表数据库表定义

登录用户信息表主要用于保存登陆时的账号、密码和时间戳。密码保存使用加密文本，具体定义参考表 4。

表 4 登录信息表

字段	类型	功能
eid_	INTEGER	用户 ID(QEID)，由服务器定义
name	TEXT(32)	登录名
psw	TEXT(16)	密码，需要保存为密文
db_ver	INTEGER	数据库版本
timestamp	INTEGER	最后登录时间戳

flag	INTEGER	标识
------	---------	----

最近更新时间戳表用于保存各个关键数据的与服务器进行的时间戳，系统会根据该表的情况判断是否发起对应数据的同步请求。在登陆成功后会主动拉取该时间戳并与本地缓存对比，同时启动定时轮询任务更新该表。具体定义参考表 5

表 5 最近更新时间表

字段	类型	功能
eid_	INTEGER	用户 ID
co	INTEGER	企业信息最后更新时间
dept	INTEGER	企业组织架构最后更新时间
emps	INTEGER	企业员工列表最后更新时间
dept_emps	INTEGER	部门与员工关系最后更新时间
vgts	INTEGER	固定组最后更新时间
default_com_contact	INTEGER	缺省常用联系人更新时间
com_contact	INTEGER	常用联系人
com_depart	INTEGER	常用部门
attention_emp	INTEGER	关注人
self_info	INTEGER	个人信息

表 6 最近更新时间表（续）

self_headimage	INTEGER	本人头像更新时间戳
empheadimage	INTEGER	同步头像时间戳

部门表用于保存组织架构中每一个部门的属性信息，每个部门都存在唯一 ID，与用户信息表中的部门 ID 对应，具体定义参考表 6。

表 7 部门表

字段	类型	功能
did_	INTEGER	部门 ID
name	TEXT(30)	部门名称
ename	TEXT(256)	部门名称 英文
pid	INTEGER	上一级部门 ID, 0 为顶级部门

tel	TEXT(20)	部门电话
orders	INTEGER	排序序号
pinyin	TEXT	拼音，用于 UI 上按拼音排序
pinyin_s	TEXT	拼音组合，用于搜索，包括多音字
flag	INTEGER	标识，除全局标识暂外无扩展
coid	INTEGER	公司 ID，暂时未使用。为与旧版相容
dept_permission	INTEGER	部门权限字段，是否显示部门，默认为 0，正常显示

员工信息表用于保存每一个员工的详细信息，会在组织架构中的名片中展示。具体定义参考表 7。

表 8 员工信息表

字段	类型	功能
eid_	INTEGER	员工 ID
ucode	TEXT(16)	员工编号(WD 为登录账号)。
name	TEXT(32)	姓名
ename	TEXT(256)	英文名
pinyin	TEXT	拼音
pinyin_s	TEXT	拼音组合，用于搜索，包括多音字

表 9 员工信息表（续）

birthday	INTEGER	出生日期
sex	INTEGER	性别
duty	TEXT(32)	职务
email	TEXT(32)	电子邮箱
tel	TEXT(20)	办公电话
home_tel	TEXT(20)	宅电
phone	TEXT(11)	手机
logo	TEXT(6)	头像
sign	TEXT(46)	个性签名
flag	INTEGER	标识

address	TEXT(80)	地址
postalcode	TEXT(8)	邮编
logoflag	INTEGER	头像更新 0 未下载 1 已下载同步 2 需要下载未同步

常用联系人表保存着员工最近联系的员工 ID 以及用户手动添加关注的人员信息，会在主界面的分组信息中展示。具体定义参考表 8。

表 10 常用联系人

字段	类型	功能
eid	INTEGER	用户 ID
is_default	INTEGER	表示该项是否是默认的联系入
flag_update	INTEGER	更新标示

常用部门表保存着员工手动添加关注的分组信息，会在主界面的分组信息中展示。具体定义参考表 9。

表 11 常用部门

字段	类型	功能
did	INTEGER	常用部门 Id
eid	INTEGER	
flag_update	INTEGER	更新标示

会话表保存员工的每一个会话的信息，包括单人聊天、多人聊天、群聊天等。该会话 ID 由服务器创建，作为唯一索引。具体定义参考表 10。

表 12 会话表

字段	类型	功能
sid_	BIGINT	会话 ID
type	INTEGER	会话类型
title	TEXT(50)	会话主题
remark	TEXT(50)	备注
flag	INTEGER	标识
cid	INTEGER	创建者 ID
ctime	INTEGER	会话创建或最后成员变更时间戳

utime	INTEGER	消息最后更新时间
off_msg	INTEGER	离线消息条数
new_msg	INTEGER	未读消息条数

日程信息表保存用户自定义的提醒事件。具体定义参考表 11。

表 13 日程信息表

字段	类型	功能
scid_	BIGINT	日程 ID
sid	BIGINT	会话 ID
eid	INTEGER	创建者 ID
name	TEXT(60)	名称
detail	TEXT(300)	内容
beginntime	INTEGER	开始时间
endtime	INTEGER	结束时间
remind	INTEGER	提醒类型
user_num	INTEGER	成员个数
users_id	TEXT	成员 ID
flag	INTEGER	标识，除全局标识暂外无扩展

3.6. 关键功能设计

本章节对系统架构中的关键功能模块的设计进行分析和讨论，主要包括基础、主窗口、会话窗口、消息管理器设置、消息提醒框、端通讯录库的加密流程等几个模块进行举例研究。

3.6.1. 基础功能

本章节从一个基本的 C/S 软件的功能组成角度出发，详细的分析了一个 windows 软件的应用基础功能

3.6.1.1. 关键界面 UI 设计

软件的基础界面主要由登录界面、主界面、会话界面、消息管理器和设置界面，其

中最核心的功能最多的界面 UI 集中在主界面（如图 5 所示）、会话界面（如图 6 所示）和消息管理器界面（如图 7 所示）

主界面中上中下分栏模式，最上方是菜单栏，除基本文件和联系人选择操作外，主要提供用户手动同步组织架构的功能；中间部分采用分栏的形式，树形图结构显示用户分组，同时用户可以选择并打开会话界面。

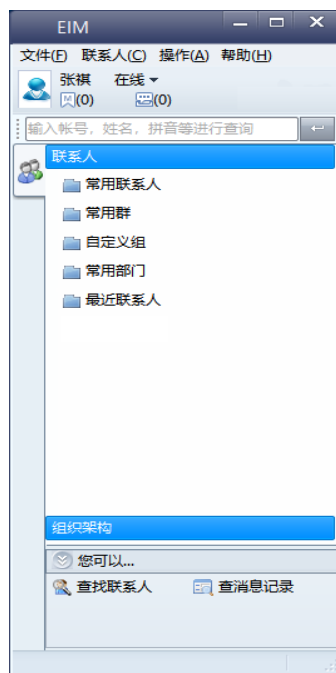


图 5 主界面

会话界面采用左中右分栏模式，左侧边栏用于显示当前所有打开的会话的标签，用户可以选择并切换焦点窗口；中间栏上方窗口显示聊天内容，下方是输入文字的窗口；右侧边栏显示的是会话对象及自身的头像和个人信息，在群组聊天中会显示当前会话中的所有成员标签。

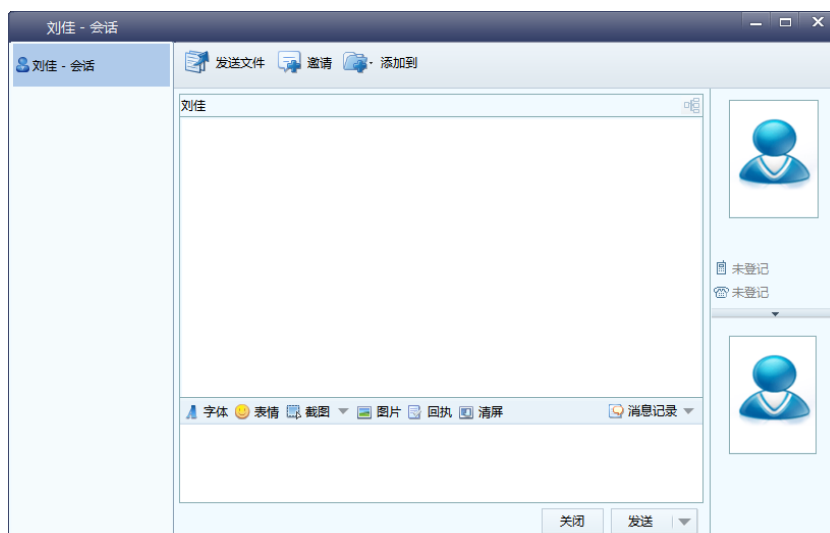


图 6 会话界面

消息管理器界面采用左右分栏模式，左侧边栏提供聊天类型的筛选，右侧边栏按照用户的输入显示查找的内容信息。

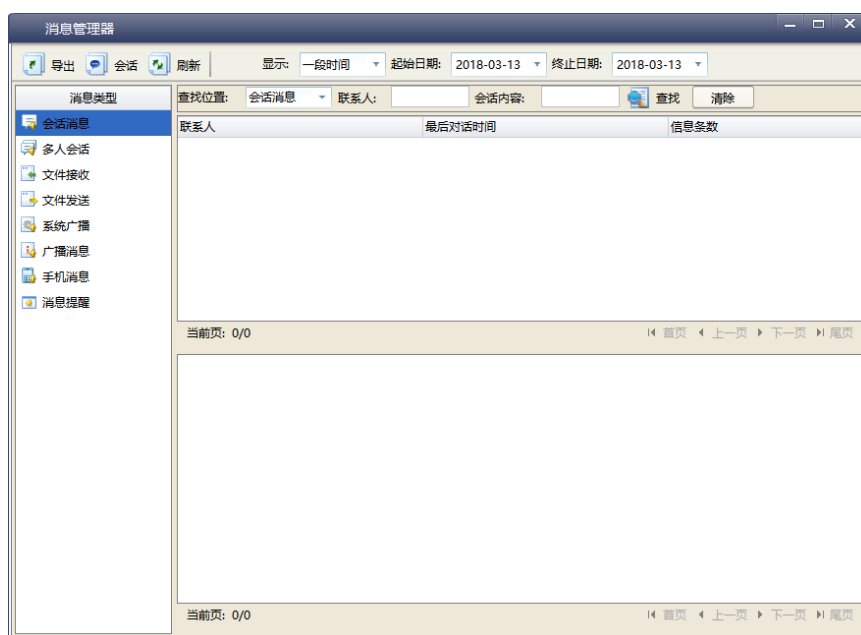


图 7 消息管理器

3.6.1.2. 软件的安装、更新和卸载

作为一个 windows 软件，首选需要实现的就是安装、更新和卸载功能。

软件安装的实现基于开源打包软件 WIX，其实现流程符合标准安全流程，根据用户选择实现注册表的写入和文件的释放。详见图 8 安装流程图。

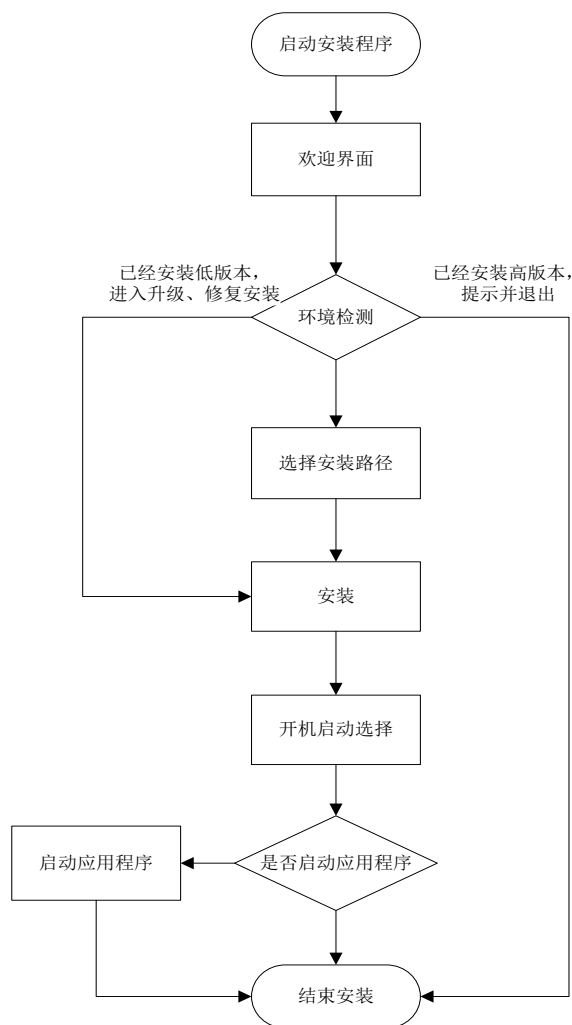


图 8 安装流程图

软件的升级方式包括静默升级和强制升级两种，系统管理员可以根据更新的紧急度在软件包的发布时进行设置。静默升级的具体功能如下：

1. 在线用户收到服务端的静默升级推送；
2. 客户端在后台进行升级包的下载及静默安装
3. 如果在升级包下载或安装过程中程序退出，不影响客户端的正常使用，在下次登录后重新下载或安装；
4. 静默安装完成后，在用户下次启动程序时弹出提示框(强制升级后不弹出此提示)：
5. 升级过程应保证原有个性化信息不会丢失；
6. 升级成功后，更新客户端本地版本信息。

强制升级的具体功能如下：

1. 用户启动程序点击登录时触发服务器的强制升级机制或在线状态收到服务器下发的升级推送；
2. 弹出升级提醒框；
3. 升级提醒信息；
4. 立即更新按钮；
5. 窗口右上角关闭按钮；

软件的升级会在用户登录的时候获取服务器最新客户端版本信息并判断是否需要升级，当需要升级时展示对应界面，界面中包含了对应的版本描述和手动下载的 URL 链接地址。用户如不进行升级，则每次登录时触发升级提醒，且不可登录。参考图 9 升级流程图。

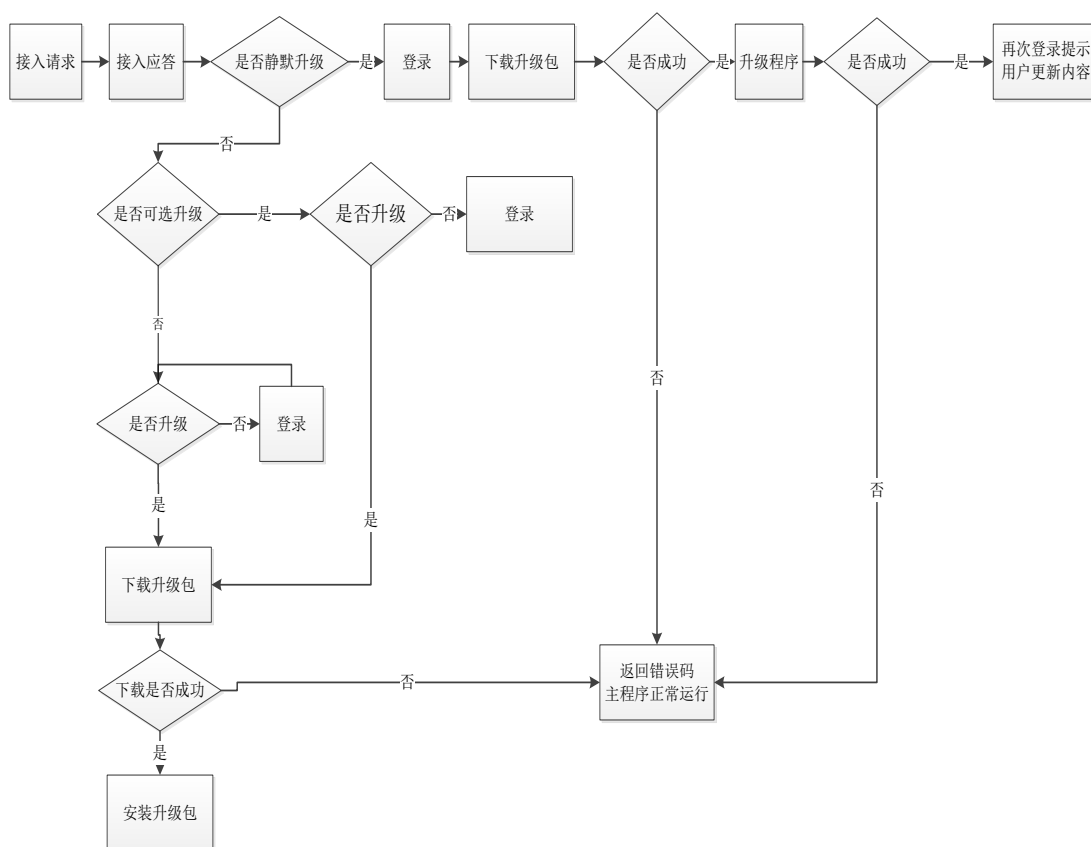


图 9 升级流程图

软件的卸载的实现基于开源打包软件 WIX，用户卸载时会先检查软件是否运行，当软件正在运行时将会提示关闭，并随后进行卸载清理注册表。

3.6.1.3. 登录和退出

用户在启动软件以后，需要输入账号密码并点击登录按钮。登录过程中会对账号信

息和客户端信息进行校验，同时进行组织架构的同步操作。具体流程见图 10。

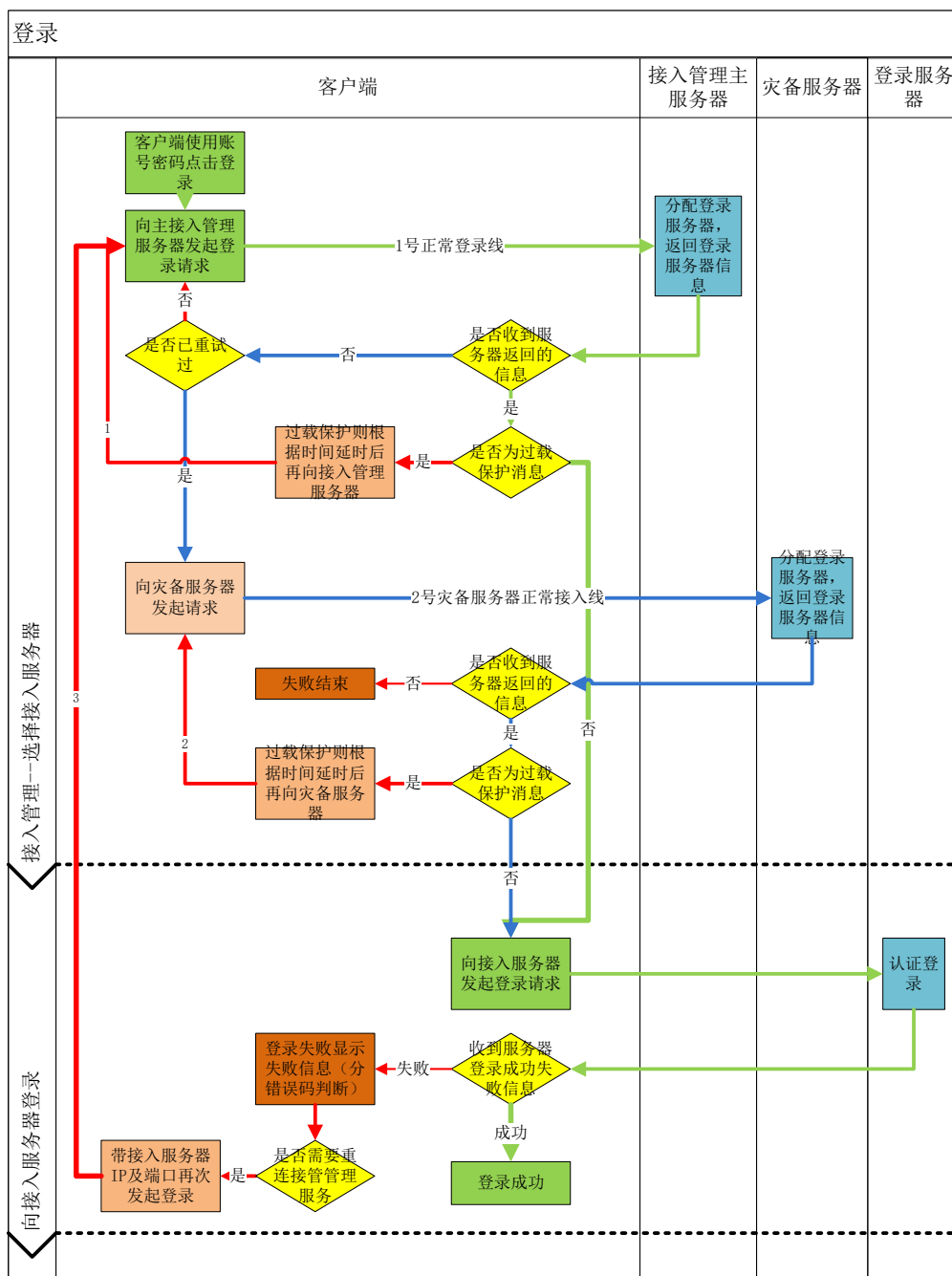


图 10 登陆、退出流程图

软件退出支持托盘右键菜单退出、主界面关闭按钮、主界面菜单退出，同时向服务端发送断开连接请求，随后结束自身运行。

3.6.1.4. 登录初始消息订阅

登录成功后，服务端会根据订阅状态进行离线消息和离线事件的推送，该操作由客

户端主动订阅，成功订阅后会收到消息推送通知，参考流程图 11。

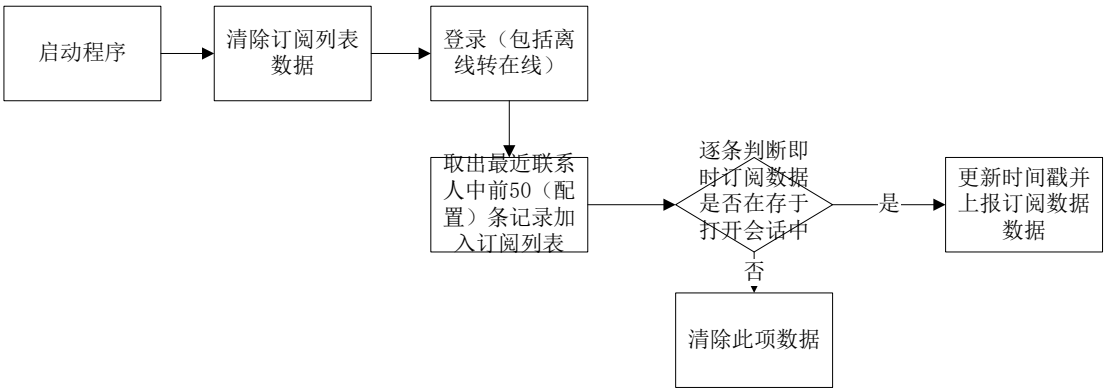


图 11 消息订阅流程图

3.6.1.5. 临时消息订阅

临时消息订阅是指当用户打开各种会话时会触发的事件。其中，会话类型分为单人会话、多人会话、群会话变更等，每种会话类型在创建会话时流程有所不同，但后续会话消息的处理流程相同。

1. 用户在组织架构中，可以选择聊天对象并打开单人会话的流程图，参见图 12

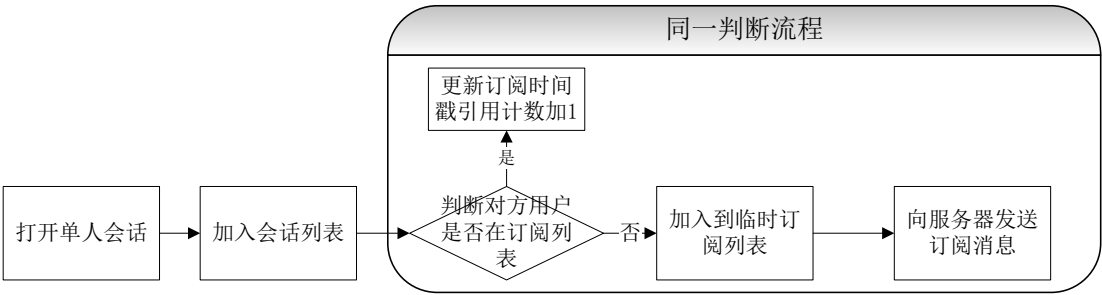


图 12 单人会话流程图

2. 用户在组织架构或自定义分组中打开多人会话，参见图 13。

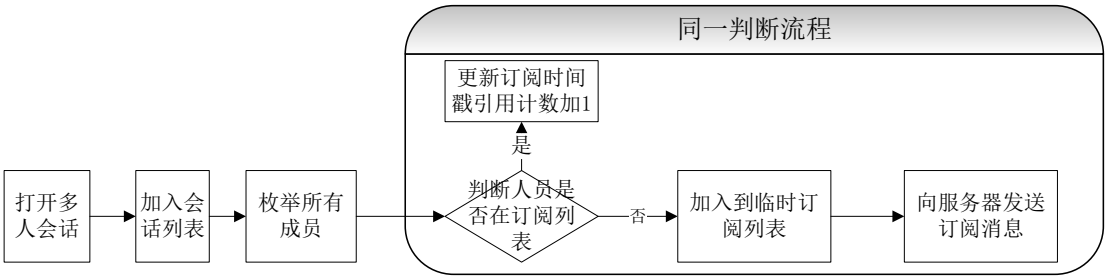


图 13 多人会话流程图

3. 用户在组织架构打开固定群会话，参见图 14。

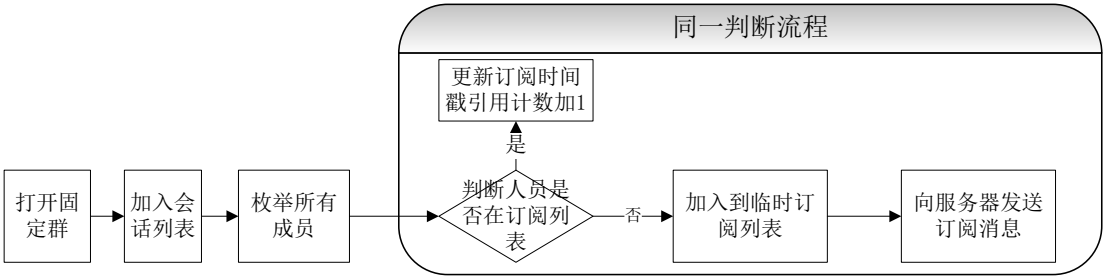


图 14 固定群会话流程图

4. 会话中的成员发生变更，参见图 15。

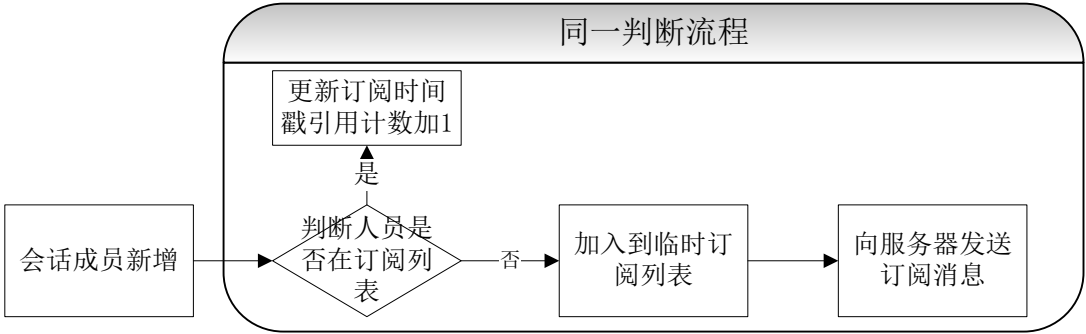


图 15 会话成员变更流程图

5. 关闭会话

关闭任何会话时只会将对应会话从会话窗口列表中删除，对会话中用户从订阅列表中进行引用计数减 1。

6. 订阅数据、会话删除流程参见图 16.

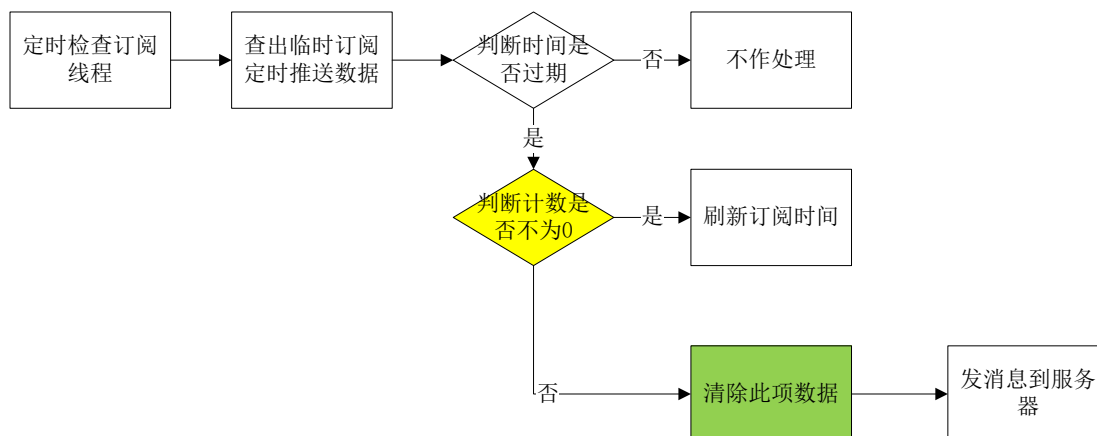


图 16 订阅流程图

3.6.1.6. 临时拉取

临时拉取指当用户需要获取某个用户、部门、组成员信息的时候，会触发的流程。同时包括某些需要定时轮询拉取的信息，比如当未触发主动临时拉取请求的时候，软件会每 12 小时自动更新一次组织架构信息。

1. 展开、收缩组织架构（常用部门）时的流程参考图 17

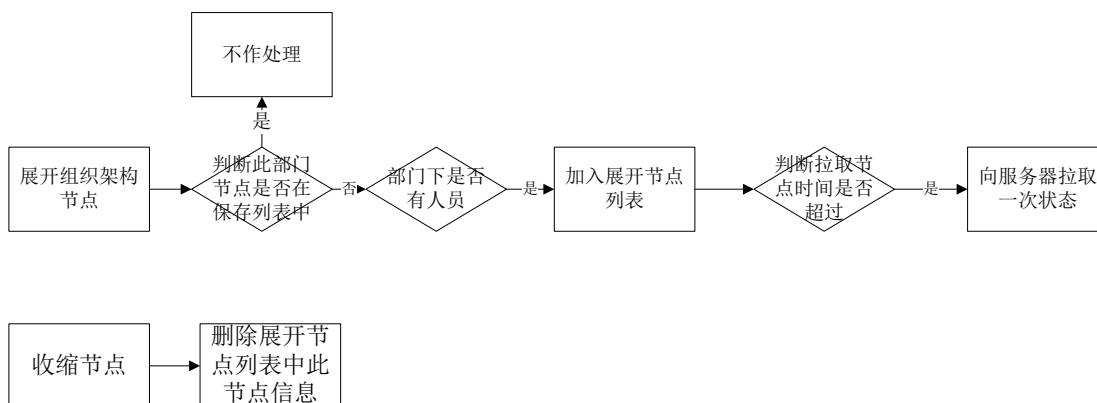


图 17 组织架构更新流程图

2. 展开、收缩自定义组（人员）时的流程参考图 18

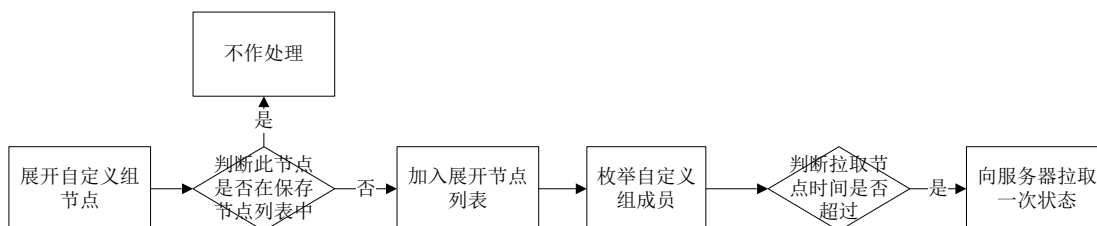


图 18 自定义组更新流程图

3. 展开、收缩最近联系人的多人会话（人员）时的流程参考图 19

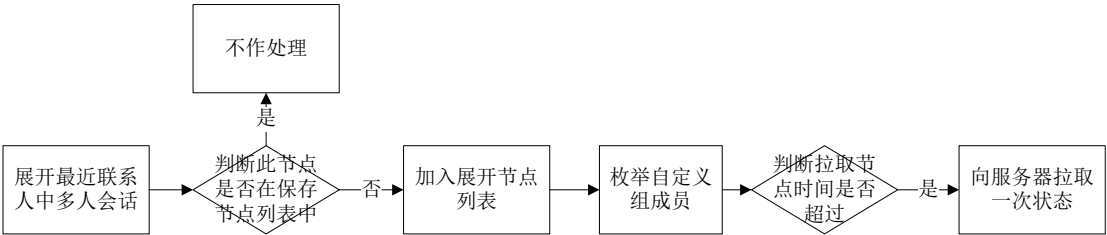


图 19 最近会话更新流程图

4. 定时拉取线程的工作流程参考图 20。

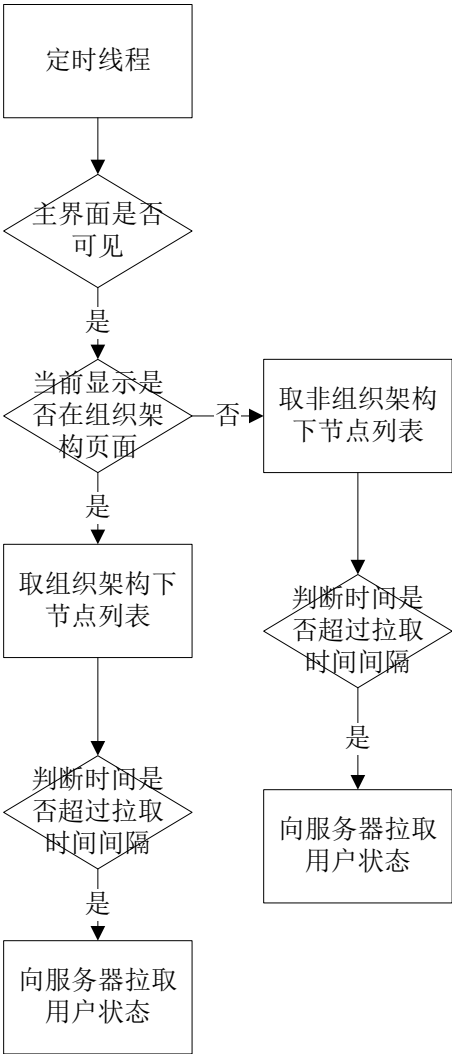


图 20 会话定时更新流程图

3.6.1.7. 用户在线状态

随着时间的推移，用户状态也会时时变化，服务器会定时将用户状态推送给客户端，

客户端在收到消息后，更新对应人员的界面 UI（比如头像），流程参考图 21。

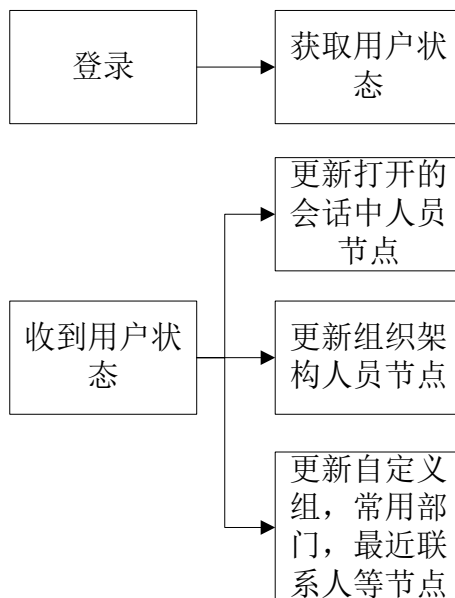


图 21 在线状态更新流程图

3.6.1.8. 搜索功能

主界面中提供搜索功能，可以搜索用户信息、部门信息和会话信息，相关操作内容和方式如下所述。

1. 当程序主界面处于激活状态时(处于桌面最顶层)，键盘输入焦点位于搜索栏；
2. 用户通过键盘使用搜索热键时（连接两次 alt）激活客户端到桌面最顶层；
3. 键盘输入焦点位于搜索栏，当搜索栏为空，或无效关键字时，enter 键激活联系人选择窗口；
4. 搜索栏可以提示用户可输入帐号、姓名中文、姓名全拼及英文名进行查询；
5. 支持即时搜，即用户输入第一个字符时即开始进行搜索并反馈搜索结果，随着用户输入的字符的变化，即时搜索并刷新搜索结果；
6. 取搜索关键字时自动过滤关键字字符串前后的空格；
7. 支持粘贴文本到搜索栏，支持粘贴从 word、ppt 中拷贝的文本，支持粘贴从 word 表格、excel 表格(多格)中拷贝的文本；
8. 搜索结果：先匹配姓名再匹配账号，全匹配优先排序，平等匹配部分按字母升序排；
9. 搜索结果展示账号(姓名)、所属部门路径，鼠标停留出现悬浮框:姓名、电话、

手机、职务；

10. 搜索结果栏支持手动拉宽；
11. 搜索栏支持搜索结果标签粘贴，即当用户从搜索结果中选中某个用户时(鼠标单击或通过键盘上、下键选中)，则该用户作为一个搜索结果标签将被粘贴到搜索栏中，搜索结果标签粘贴到搜索栏中后，不影响用户再次输入关键字搜索；
12. 搜索结果中的第一个需默认选中状态；
13. 搜索栏中标签的键盘响应：当搜索栏中具有标签时，键盘左右键位移粒度为一个标签，上下键位移粒度为一个字符；
14. 键盘输入焦点位于搜索栏，且搜索栏中具有标签，用户按回车键时，该动作将打开搜索栏中所包含的所有搜索结果标签所指向的人的多人会话，包括此时下拉搜索栏中所默认选中的搜索结果（如有，此是相应 **enter** 的操作时先将下拉搜索栏中默认选中的结果作为标签粘贴到搜索栏，再激活包含搜索栏所有标签的会话）；
15. 当搜索栏中具有搜索结果标签时，如用户在标签字符串范围内单击鼠标左键，则选中标签；双击时选中所有标签；
16. 搜索栏支持右键菜单，包含“撤销、剪切、复制、粘贴、删除、全选”，当搜索栏为空时，剪切、复制、删除、全选选项需为不可选状态，撤销是指撤销前一个搜索栏操作（包括删除、复制、粘贴、剪切）；

3.6.1.9. 自动重连

当网络发生异常波动时，导致心跳中断，服务端将当前 ID 在线状态重置。若此时产生了订阅请求消息等非登录认证的消息类型时，会触发自动重连的逻辑以更新和同步

最新用户或服务端的状态，具体流程参考图 22。

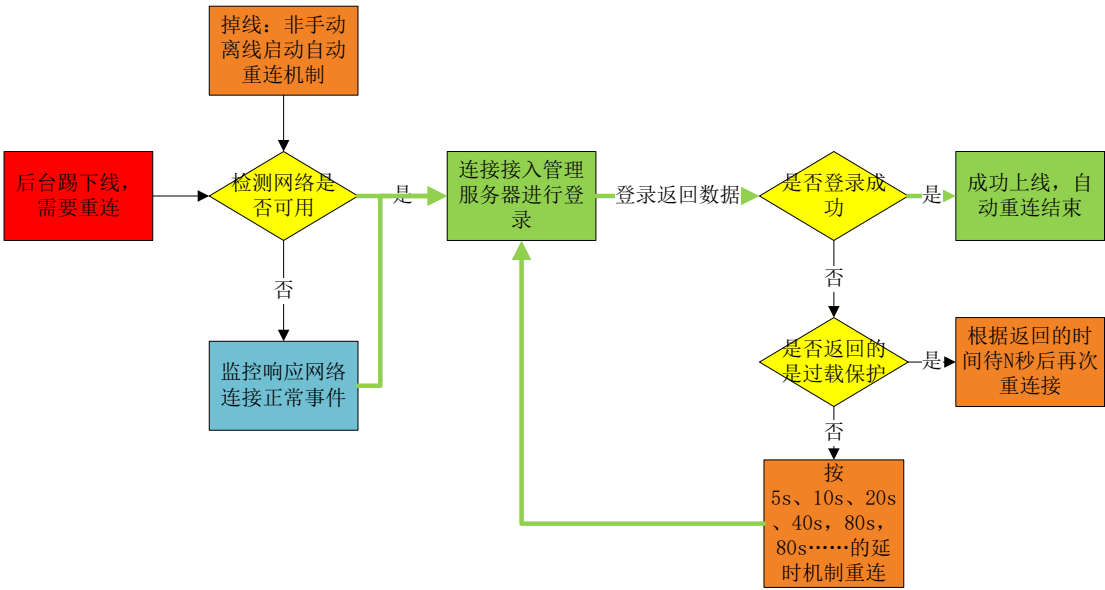


图 22 自动重连流程图

3.6.2.主要功能

3.6.2.1. 个人设置

个人设置界面允许用户输入自身基本信息和联系方式，同时可以设定头像。点击头像可以进入设置界面，具体位置定位到个人设置中的基本设置。

头像的如果设置了个性头像，会在登录时从服务器下载头像然后显示，否则显示默认头像（包括默认男头像和女头像）。

3.6.2.2. 登录状态变更

用户可以在主界面个人信息栏中或托盘图标栏中定义自身的在线状态，包括在线、离线和离开三种状态，其中离开状态可以设定自动回复的文本，状态变更的流程参考图 23。

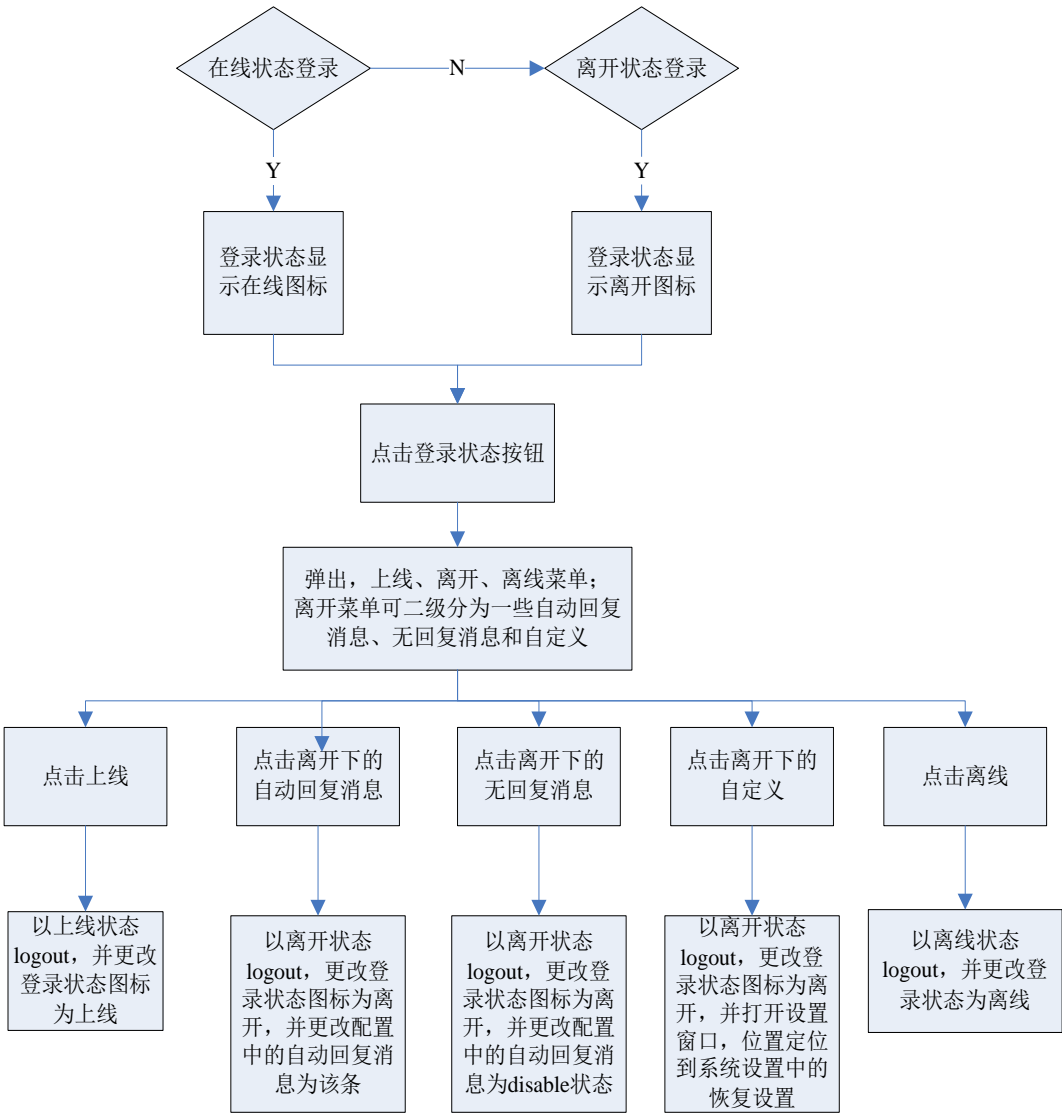


图 23 登录状态变更流程图

3.6.2.3. 查看未读（离线）消息

当发来新消息时，应给予用户通知，需要在已打开的会话中显示或者在托盘区闪烁，获取未读消息的流程参考图 24

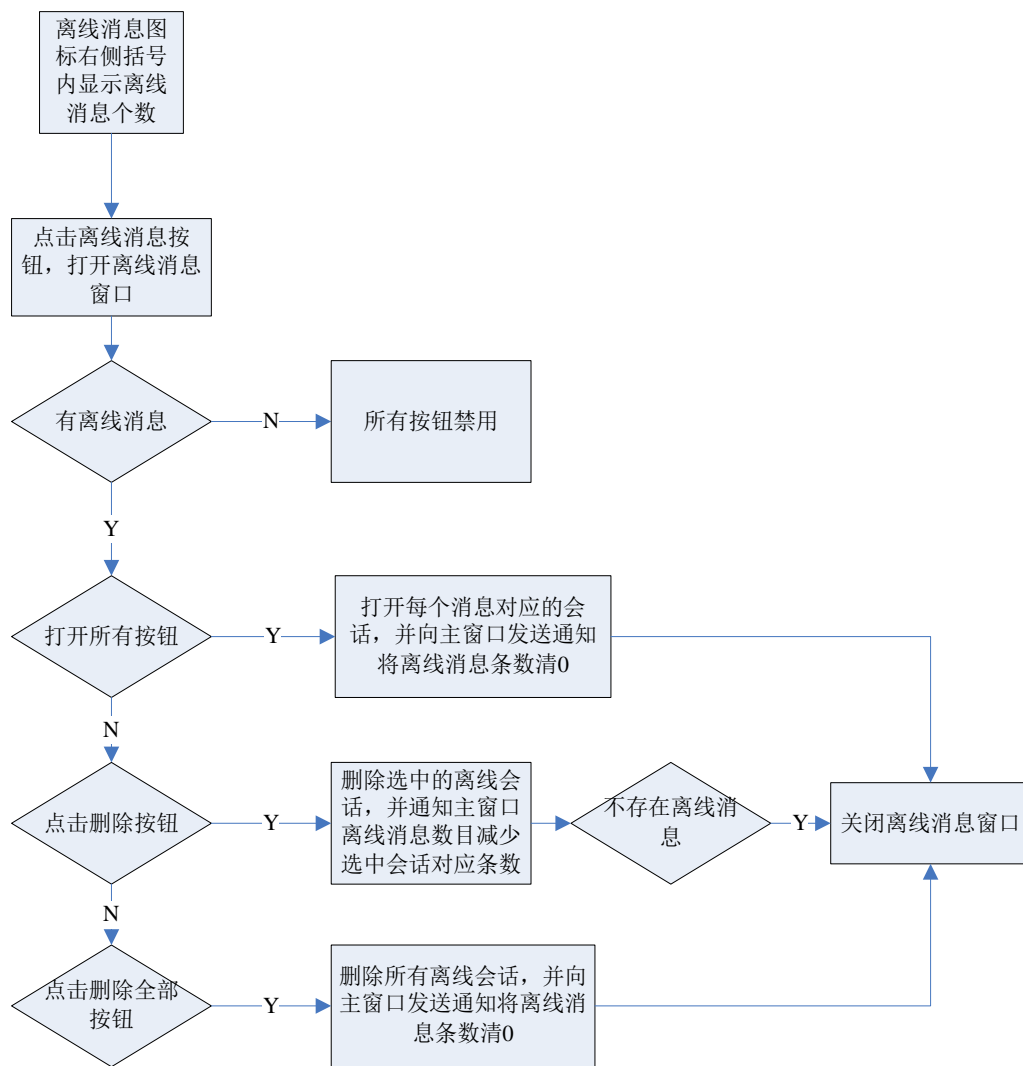


图 24 更新消息流程图

3.6.2.4. 快速查找

参照 3.2.1.8 搜索模块。

3.6.2.5. 组织架构面板

3.6.2.5.1. 组织架构

组织架构的显示是一个树型结构，它的结构包括部门、人员。部门要具有折叠的功能，人员项不具有折叠功能。

折叠功能的实现是利用列表项的隐藏、显示。

组织架构会定时与服务器进行同步，也会在用户打开时任何组信息时进行更新，用户也可以手动更新。具体的更新流程参见图 25。

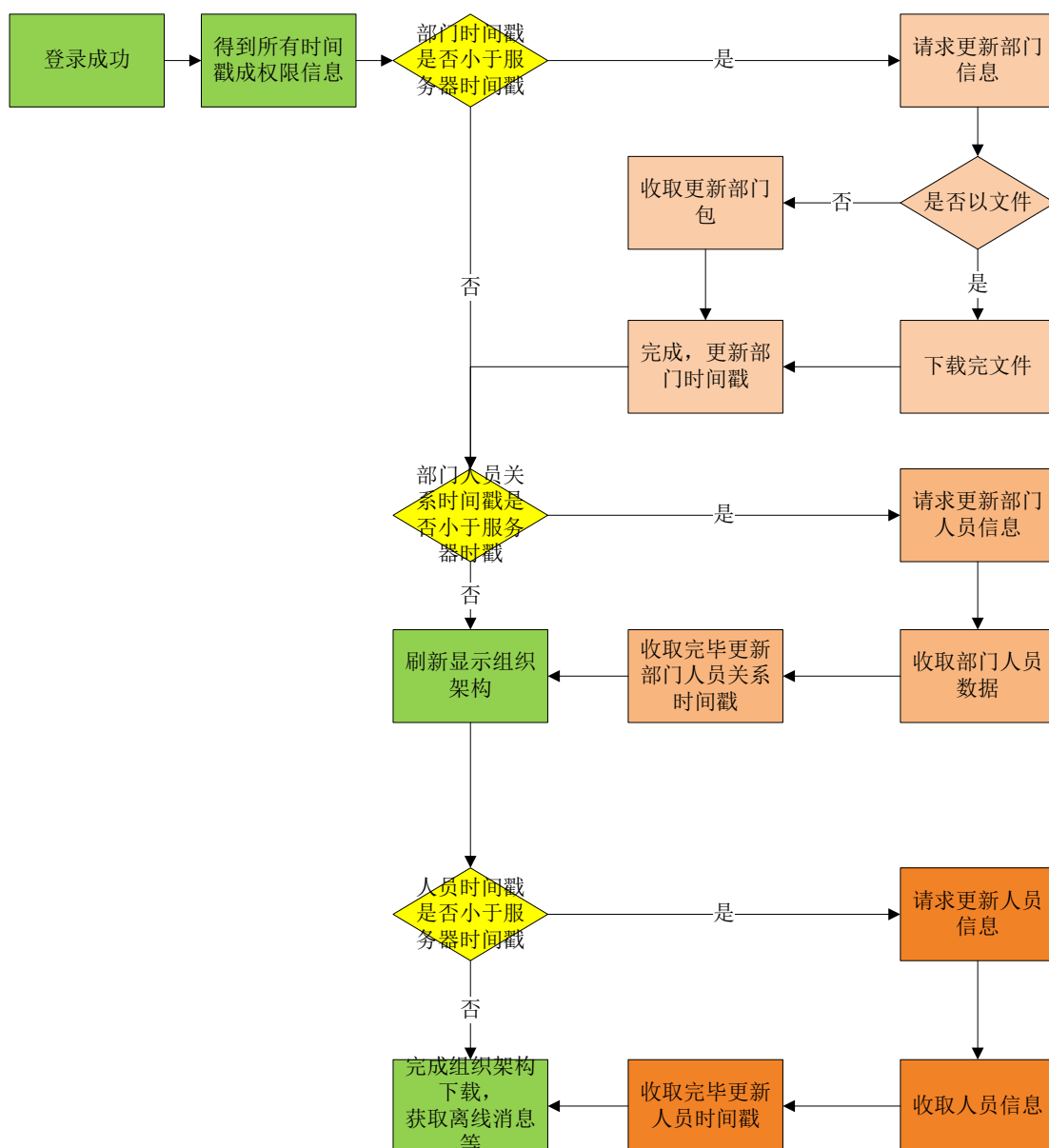


图 25 组织架构更新流程图

3.6.2.5.2. 部门

在组织架构选项卡中，支持按部门展示架构。

1. 添加到常用部门：

用户可以在部门中右键选择并添加到常用部门。

2. 在线人员列表：

点击在线人员列表，可查看到该部门当前在线人员。通过获取当前部门 ID 通过引擎枚举出在线的人员。

3.6.2.5.3. 员工

用户在员工选项卡上可以双击打开聊天窗口，或右键调出菜单栏并进行对应的操作，具体内容如下：

1. 发送即时消息，基本流程参考图 26。

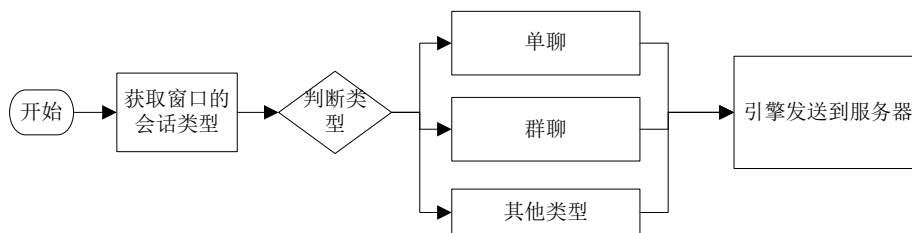


图 26 消息发送流程图

2. 发送邮件。

3. 发送广播消息。

4. 发送文件：如果选择的单人则启动单人会话，选择了多人则发起多人会话。并启动发送文件选择框。

5. 添加到常用联系人：参考 3.2.2.6.4 联系人项。

6. 查看消息记录：点击消息记录按钮可查看消息记录。利用当前会话的 ID，通过引擎枚举出该会话 ID 的所有会话信息。

7. 查看联系人信息：可查看联系人的详细信息，如电话、姓名、工号等等。利用当前会话接收方 ID,向引擎索取改 ID 人员的详细信息。界面利用获取到的数据进行显示。

8. 关注联系人状态：可关注联系人，组织架构人员项上显示出取消关注的按钮。

3.6.2.5.4. 联系人

联系人界面采用树形图结构展示，支持折叠展开，信息可增删改操作

这个树形结构里面的项可分为：部门项、人员项、文件加项、群组项。树的实现类似于组织架构。

联系人的数据展示内容保存在 XML 格式的配置文件里面，启动的时候加载数据，生成树型结构。修改添加时修改数据库。

3.6.2.5.5. 用户头像的下载与更新

头像用于展示员工个性照片，会在订阅员工个人信息时进行下载，不会一次性下载全部员工的头像。

头像的时间戳数据保存在员工信息表中，每个员工在头像未下载时均使用默认的头像（男和女两种）。

下载地址为服务器配置的地址接用户 RTXID 模式来下载，在收到服务器通知之后启动文件的下载。

图片的大小固定为 84*112，按此长宽比例进行缩放。

3.6.2.6. 组织架构页面

登录成功后，就加载顶层部门。当展开部门时，再加载该部门下面的人员与部门子节点，自动订阅该部门下面的成员状态，折叠时取消订阅。

3.6.3. 会话窗口

3.6.3.1. 会话管理器

会话管理器模块负责监控、保存每一个会话信息，同时处理界面的 UI 消息。该管理器设计原理如下：

1. 会话窗口管理、会话窗口创建都是通过会话管理模块实现。
2. 会话管理模块保存一个会话 ID 和 会话窗口的一个映射。
3. 会话管理的基本逻辑参考图 27。

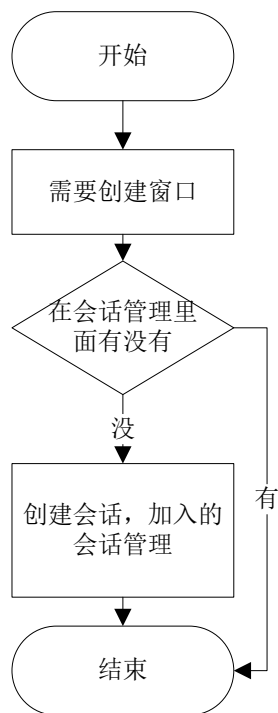


图 27 会话管理流程图

4. 在创建会话窗口的时候，进行人员个数、或权限的验证，如果验证通过则创建会话窗口。

3.6.3.2. 主菜单

主菜单模块中实现对快捷键的支持并实现对应功能，快捷键的设置设置在设置界面可以自定义，默认的快捷键及功能如下：

1. 支持 Ctrl+Z 撤消。
2. 支持 Ctrl+X 剪切。
3. 支持 Ctrl+C 拷贝 和右键复制拷贝。
4. 支持 Ctrl+V 粘贴 和右键粘贴。
5. 支持 Ctrl+A 粘贴 和右键全选。
6. 支持点击清屏按钮清屏，和右键清屏。
7. 通过更改字体选项 设置更改发送的字体。

3.6.3.2.1. 发送文件

用户在聊天窗口中可以通过工具栏发送文件按钮唤出文件选择窗口并发送文件，同时也支持将文件拖拽至聊天窗口中触发发送文件的操作，具体流程参考图 28。

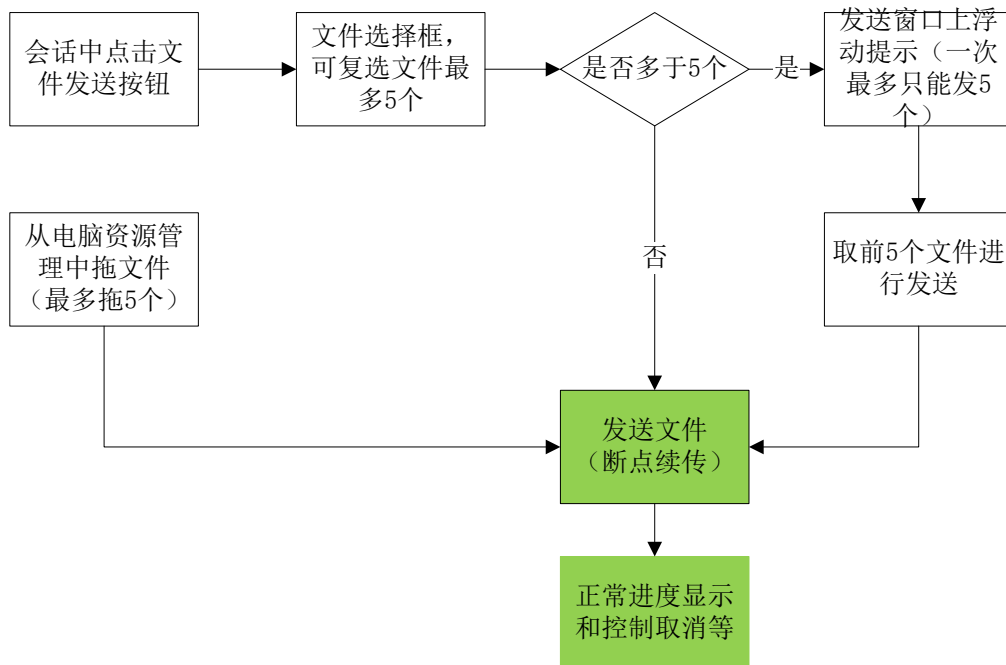


图 28 文件发送流程图

3.6.3.2.2. 捕捉屏幕

程序内置截屏软件，实现原理为在桌面上方创建灰色图层，获取用户选择的区域进行透明化处理，并保存对应桌面截图。

当用户点击该功能时，通过 `ShellExecute` 函数唤起新进程，在用户完成截图后将该图片自动复制到聊天窗口中。

3.6.3.2.3. 选项

1. 窗口置顶：

勾选该选项可将当前窗口置于屏幕顶层，去选该选项可取消窗口置顶；

窗口置顶选项只对本次打开的本窗口有效，即当该窗口关闭后，置顶选项将被重置(重新打开后窗口依然默认为不置顶)；

2. 侧边栏：

勾选可打开侧边栏，去选可关闭侧边栏，该选项默认为勾选(默认打开侧边栏)；

对于单人会话，侧边栏显示本人头像，对方头像及对方联系电话(手机、办公电话、电子邮件)；该选项为全局配置选项(是一个全局配置功能，具有记忆)；

对于多人会话，侧边栏显示参与人列表；该选项为本会话本次打开窗口选项，窗口关闭后则重置；

3. 自己头像：

勾选可在侧边栏展示自己头像，去选在侧边栏不展示自己头像，该选项默认为勾选；全局配置选项(是一个全局配置功能，具有记忆)；

4. Enter 或 Ctrl + Enter 发送：

这 2 个选项是互斥选项、全局配置选项，用户可以设置快捷键用于发送聊天内容。

3.6.3.3. 消息发送框

3.6.3.3.1. 字体

对话框创建时，加载字体。字体通过 XML 格式保存。支持用户手动选取字体，字体属性保存在数据库，并通知输入窗口。

3.6.3.3.2. 表情

表情的本质是图片，资源文件存放在本地。用户发出“表情”时会自动转换成 XML 格式文本，接收方解析文本后读取本地表情图片并显示在界面中

3.6.3.3.3. 图片

支持拖拽图片和用户手动选择，具体实现由 RichEdit2.0 模块实现。

3.6.3.4. 回执

回执属性是一种需要消息接收方做出回应操作的功能，接收方需要点击“发送回执”按钮完成回执的发送，通知消息发送者你已收到知晓该消息。具体流程参考图 29。

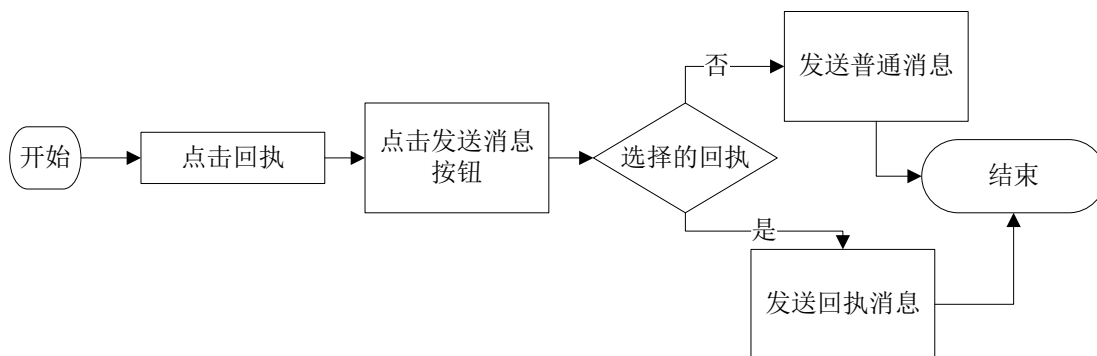


图 29 更新回执流程图

3.6.3.5. 群聊屏蔽

群聊可屏蔽消息提示。如果设置了“接收不提示”，消息只做接收处理，不触发消息提示。

3.6.3.6. 会话成员信息

对话窗口打开时加载了成员信息，成员信息在窗口右侧边栏显示。

3.6.3.7. 邀请

点击邀请按钮，弹出选择人员窗口。窗口中展示默认组织架构，将选择的人员加入到聊天会话中。如果是单人聊天聊会话通过邀请会自动新建多人聊天会话，如果是多聊通过邀请可增加人员。

3.6.4.消息管理器

1. 消息查看类型：

会话消息、多人会话、文件接收、文件发送、系统广播、广播消息，手机消息，消息提醒。

2. 消息查看时间段包括全部、最近七天、一段时间（手动输入）

3. 搜索条件支持如下：

会话记录：联系人、会话内容；

文件记录：发送人、文件名；接收人、文件名；

广播：广播内容；

手机消息：联系人、短信内容；

消息提醒：主题、内容；

4. 消息记录导出，导出数据库文件以 Excel 格式存储；

5. 消息记录导入

支持导入消息记录，导入消息记录根据会话 ID 匹配插入，如原记录不存在的会话 ID，则为新会话记录；

6. 消息框以分页形式显示

3.6.5. 设置

3.6.5.1. 个人设置

个人设置界面主要提供对用户基本资料的修改，但是不允许用户修改自身 ID。在发生修改后会与服务器进行更细，根据回应包同步到本地数据库。

用户可以修改自己账号的登陆密码以及自身的头像，头像修改成功后由客户端记录更新时间戳。

个人设置界面发生修改后用户通过单击“确定”按钮提交更新，具体更新流程参见图 30。

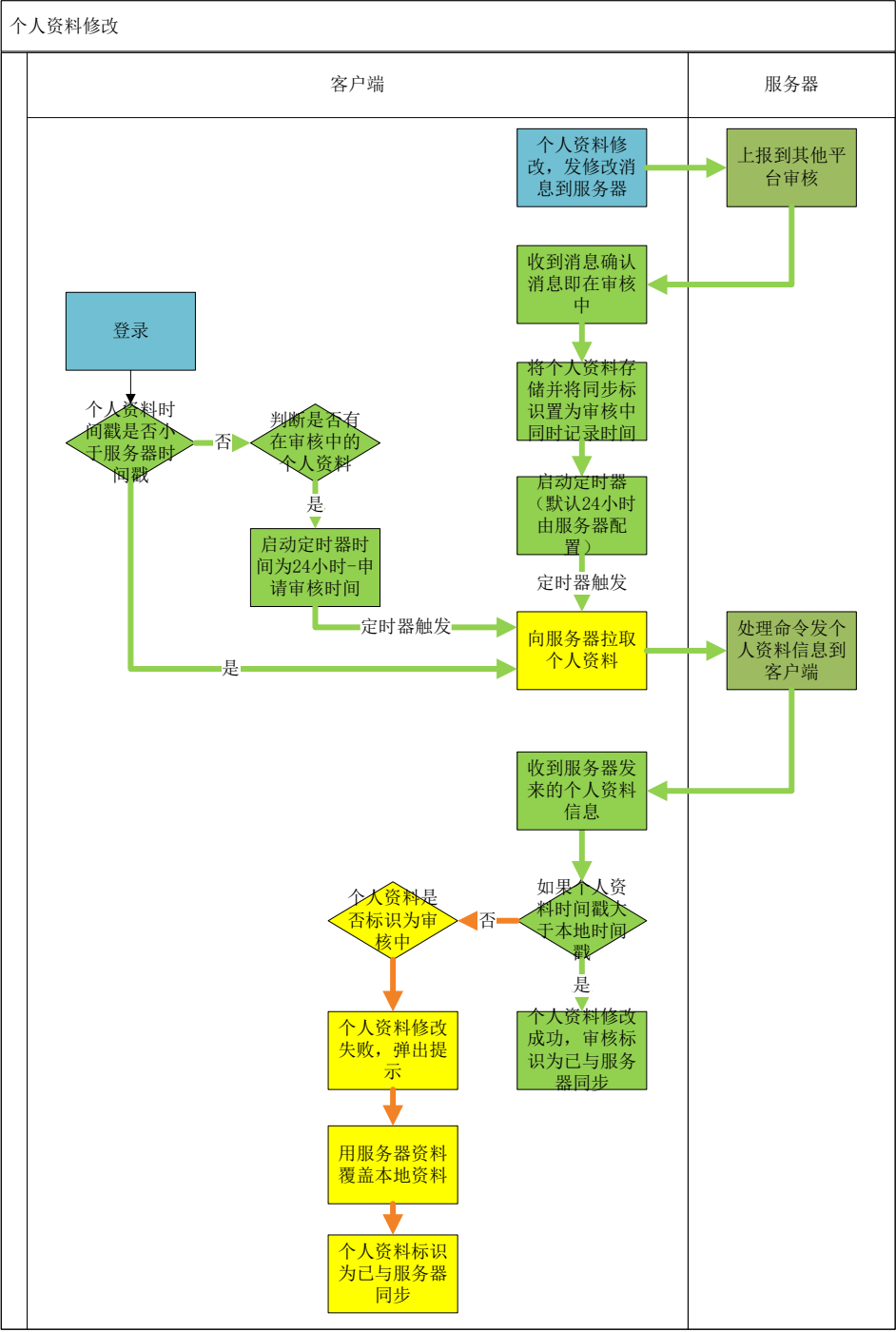


图 30 个人设置更新流程图

3.6.5.2. 系统设置

区别于个人设置功能需要和服务器进行通信，系统设置功能主要针对软件运行时的状态设置，还包括网络连接代理和热键设置，这些功能主要涉及模块间的消息传递。同时用户可以自定义“自动回复”消息的内容，所有的设定会同步到本地数据库保存。

3.6.6.消息提醒框

用户在接受到新消息时，若对应的会话界面没有被创建或打开，需要以提示窗口的形式弹出，参考 QQ 的消息提醒功能，软件会在桌面右下角创建对应窗口，窗口中会显示消息内容。同时支持用户单击该窗口后创建或者打开对应会话的功能。

3.6.7.托盘

程序运行后在系统托盘区具有程序图标，且可切换变换不同状态、不同消息通知的图标，鼠标移动到图标上弹出“Tips”显示当前登录帐号信息，并提供鼠标右键菜单栏，菜单栏的具体功能包括个人设置(离线状态为不可选)、系统设置、上线、离开、离线、显示主界面、更改用户、退出。

3.6.8.端通讯录库的加密流程

数据库是保存所有聊天记录的储存介质，是重要的个人隐私，因此需要进行加密处理，防止个人资料的泄露或者用户本人修改数据库导致程序运行异常。

确认 SQLite 是已经支持数据库加密的版本，如果本地没有通讯录，会在下次登录时进行下载；如果本地通讯录没有加密，就会在下次对数据库进行 OPEN 操作时进行加密处理，具体的加密流程参见图 31。

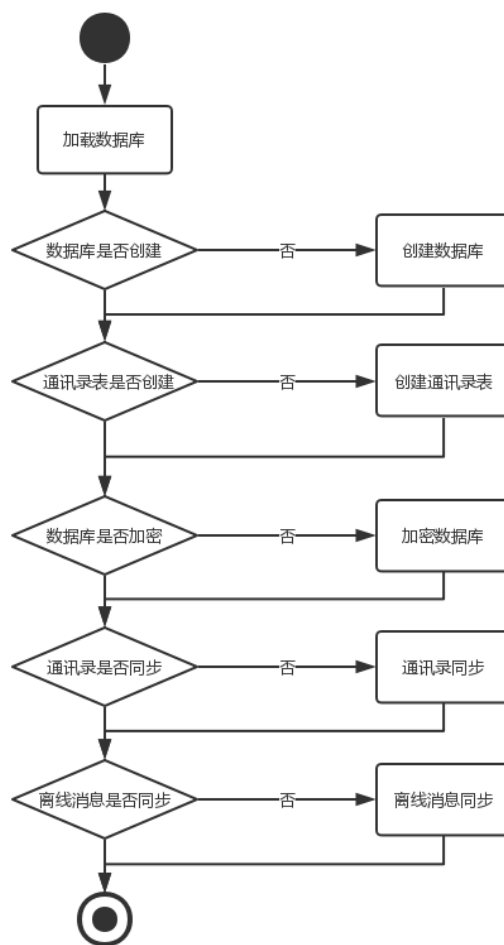


图 31 数据库加密流程图

数据库访问需要遵循以下几点原则：

1. 由于 SQLite 数据库设置密码时，必须是空的库。所以，如果已经存在数据的库需要加密，就会创建新的库，设置密码后，再导入旧的数据。
2. 如果已经使用了支持加密版本和程序，数据库就会被加密，无法倒回去使用未加密的版本程序（包括全局库 `users.db`，它可能会造成一个假象：登录密码无法保存）
3. 已经加密的库，就会直接访问。

3.7. 满足系统性能需求的设计

在本章的第 4 节对系统性能需求的设计进行了介绍，主要包括稳定性、可靠性、和安全性。

第四章 系统测试

4.1. 测试概述

系统测试是在单元测试和集成测试完成之后，对软件进行系统性测试，比如初始化测试、功能测试、数据处理测试、压力测试、性能测试等。一般情况下，系统的测试包括对软件和硬件的测试，本文仅针对软件测试进行说明和描述。

4.2. 测试目的

测试目的主要包括及时发现数据或业务逻辑缺陷并修复，保证功能完整性；以及稳定性和性能瓶颈的优化以提升用户体验。

通过系统测试，确认软件是否达到了预期功能要求，纠正系统设计时存在的逻辑错误或者代码设计人员在逻辑控制上的失误等造成的一系列问题。

4.3. 测试工具和测试环境

基于 C/S 架构，分别区分为服务端环境和客户端环境，其中常用的测试工具主要包括网络通信抓包工具 Wireshack 和网络接口测试工具 Postman。

4.3.1.服务端测试环境

服务端测试环境配置需求参见表 12，其中服务器环境二为备用服务器：

表 14 服务器测试环境

	服务器环境一	服务器环境二
CPU	Intel Xeon E3-1281 v3	Intel Xeon E3-1281 v3
内存	512GB	256GB
硬盘	1TB	1TB
操作系统	CentOS 7	CentOS 7
数据库配置	Oracle	Oracle

4.3.2. 客户端测试环境

客户端测试环境主要使用虚拟机配置，基本测试环境为参考表 13：

表 15 客户端测试环境

硬件设施	客户端环境
CPU	i5-5200U@2.20GHz
内存	2GB
硬盘	120GB
显卡	NVIDIA GeForce GTX 470M
分辨率	1280x768

操作系统分别包括 WinXP、Win7、Win8、Win10、WinServer 2008、WinServer 2012、WinServer 2016。

4.4. 测试流程

1. 制定测试计划，根据需求分析制定测试需求报告，即黑盒测试标准。所有的测试工作均以测试需求为基准进行，以确定应用程序是否合格。
2. 制定测试方案，需求点分析，模块划分，流程图分析。
3. 编写测试用例并评测。
4. 执行测试用例，提交缺陷信息和测试结果报告。
5. 回归测试，跟踪缺陷修复情况。
6. 验收测试，出具测试评估和建议报告。
7. 测试归档。
8. 产品上线后跟踪及后续维护。

4.5. 典型的功能测试用例

功能测试，是用来检查业务逻辑是否符合需求分析，系统的实现能否够满足后续的测试需求。功能测试主要是检查输入和输出是否符合预期，校验客户端、服务器与数据库之间进行的数据交换在处理后显示的是否正确。同时检验系统的功能有没有达到基本的要求，对于没有满足功能测试的项目不应进行后续测试。

该测试在整个流程中起着关键性的作用，其目标是测试系统的功能是否与符合预期，测试的方法主要是根据需求分析中的模块一项项的进行测试，验证是否与需求是一致的。同时，测试过程中要考虑功能或需求发生了变化的情况。对于这类情况要重新进行测试，保证功能的正确执行。

由于测试用例较多，本文选取了 2 个重要测试用例作为示例。

1. 登陆界面功能测试用例，详见表 14：

表 16 登陆界面功能测试用例

用例编号	用例描述	测试数据	预期结果
TLOGIN_1	用户名正确，密码正确，点击登陆按钮。	用户名：admin 密码：admin	登陆成功
TLOGIN_2	用户名大小写混合，密码正确，点击登陆按钮。	用户名：Admin 密码：admin	提示用户登陆失败，账号或密码错误
TLOGIN_3	用户名正确，密码大小写混合，点击登陆按钮。	用户名：admin 密码：Admin	提示用户登陆失败，账号或密码错误
TLOGIN_4	用户名不存在，密码正确，点击登陆按钮。	用户名：null 密码：admin	提示用户登陆失败，账号或密码错误
TLOGIN_5	用户名正确，密码错误，点击登陆按钮。	用户名：admin 密码：null	提示用户登陆失败，账号或密码错误
TLOGIN_6	用户名为空，密码正确，点击登陆按钮。	用户名： 密码：admin	提示请输入用户名，并且光标焦点改变为用户名输入窗口
TLOGIN_7	用户名正确，密码为空，点击登陆按钮。	用户名：admin 密码：	提示请输入密码，焦点改变为密码输入窗口
TLOGIN_8	输入正确密码，复制粘贴字符串至账号输入窗口，点击登陆按钮。	字符串：admin 用户名：admin	用户名支持复制，且登陆成功

表 14 登陆界面功能测试用例（续）

TLOGIN_9	输入正确的用户名，复制字符串至密码输入窗口，最后单击登陆按钮。	用户名：admin 字符串：admin	密码支持复制，且登陆成功
TLOGIN_10	在密码窗口，输入任意字符，检查是否加密“*”		每个字符在输入后均显示为“*”
TLOGIN_11	检查焦点在用户名窗口时，输入 TAB 键，光标焦点是否跳转		光标焦点跳转到密码输入窗口
TLOGIN_12	检查焦点在密码窗口时，输入 TAB 键，光标焦点是否跳转		光标焦点跳转到用户名输入窗口
TLOGIN_13	焦点在登录窗口，按下 ENTER 键		成功触发登陆逻辑
TLOGIN_14	在密码窗口，输入任意字符，检查是否加密“*”		每个字符在输入后均显示为“*”
TLOGIN_15	点击关闭或取消按钮，检查登录窗口状态。		登录窗口成功关闭。

2. 单人聊天窗口功能测试用例，详见表 15。

表 15 单人聊天功能测试用例

用例编号	用例描述	测试数据	预期结果
TSCHART_1	编辑窗口内，输入字符串、数字	字符串：abcde_123	成功输入
TSCHART_2	网络正常，发送字符串	字符串：abcdefg	对方成功接收到该字符串
TSCHART_3	网络正常，发送图片、表情	任意 png、jpg、bmp、gif 格式图片	对方成功接收到该图片
TSCHART_4	鼠标点击截屏工具按钮		成功打开截屏工具程序
TSCHART_5	鼠标点击字体修改按钮		成功打开字体修改窗口
TSCHART_6	测试常用字体修改	字体：宋体 大小：10	确定修改后，仅编辑窗口字体改变。
TSCHART_7	测试常用字体修改	字体：微软雅黑 大小：10	确定修改后，仅编辑窗口字体改变。
TSCHART_8	测试常用字体修改	字体：隶书 大小：10	确定修改后，仅编辑窗口字体改变。
TSCHART_9	测试常用字体大小修改	字体：隶书 大小：15	确定修改后，仅编辑窗口字体大小改变。
TSCHART_10	测试常用字体大小修改	字体：隶书 大小：5	确定修改后，仅编辑窗口字体大小改变。
TSCHART_11	网络断开时，发送字符串、图片。	字符串：abcd_123 任意 png、jpg、bmp、gif 格式图片	聊天内容成功显示在聊天栏窗口，等待一段时间显示发送失败，提示重新发送。
TSCHART_12	网络断开时，发送字符串、图片失败后，点击重新发送。		一段时间显示发送失败，提示“重新发送”

4.6. 性能测试

本项目中，性能测试主要是测试与界面交互时，输入并提交相关数据后，经过系统后台服务器、数据库、网络通信等操作处理最终返回到客户端上时，所需的反应时间以及数据是否符合预期等。

测试目的是检查系统性能是否符合预期，且 CPU、内存使用正常，没有内存泄漏等问题。同时，对于特殊要求模块需要更严格的性能指标。

通过测试结果反馈，本系统性能够满足集团的基本需求，性能基本符合预期，同时需要后期持续优化，具体性能指标详见表 16。

表 176 性能指标

性能模块	性能指标
消息丢包率	<0.5%
消息接收响应时间	<=500 毫秒
CPU	<10% (Intel i7-4790 @ 3.60GHZ)
内存	<10% (标准 8G 内存)
崩溃率	<1 次 (每 100 次运行)

4.7. 测试结果分析

本即时通讯软件按照软件工程的测试方法进行了详细的测试，从系统的测试概述、测试环境、测试内容、测试分析等方面对其进行了阐述与分析，并且对于测试过程中出现和发现的问题进行了改正，测试的结果符合预期，能够满足集团的日常办公需求，具体测试摘要分析报告参见表 17。

表 187 测试分析报告

缺陷类型	产生缺陷总数	修复缺陷总数	修复率	缺陷密度	占总缺陷比率
致命缺陷	1	1	100%	0.3%	3.7%
严重缺陷	2	2	100%	0.6%	7.4%
一般性缺陷	17	17	100%	5.1%	62.96%
提示性缺陷	6	6	100%	1.8%	22.2%
性能缺陷	1	1	100%	0.3%	3.7%
合计	27	27	100%	8.1%	100%

综上所述，通过测试证明系统基本实现了预期，满足了系统设计部分所需功能，交付并成功通过测试，保证了系统的业务逻辑流程的正确性。

第五章 系统实施

5.1.实施概述

根据集团发展的需求需要在内部实施基于集团现有组织架构的企业级即时通讯软件，即本系统终端和后端服务器。系统实施需要保证目标对接系统平台的稳定和系统数据的安全加密，并在上线前完成系统的初始化，以及满足上线后的平稳运行的实施要求。

系统实施的目标是启动本系统软件的后台服务系统和消息备用系统，通过小范围的选取部门安装客户端软件进行测试，以确定实施结果可以满足预期。并最终切换至正式服务器，推广至剩余部门。最后通过本系统实施满足集团下各个子公司内部的日常沟通、提高工作效率及效益，满足信息时代集团对信息安全的需求。

本章节主要针对集团现有系统环境和服务器实施上线进行详细描述，其中客户端软件将直接进行域分发安装或上传至 FTP，通过邮件通知用户下载并安装。

5.2.实施计划

根据集团内部项目管理流程，将项目的实施分为启动、实施和验收三个阶段。对于项目中涉及的实施流程包括：项目启动、集团内部需求调研、需求分析、系统设计、详细设计、编码调试、系统集成、系统测试、系统验收、 α 测试和交付上线。

项目的实施之前，通过设立实施小组确保系统正常平稳的过渡。并在产品正式上线之前，负责实地操作和监控项目在上线前的活动。准备工作完成后将按照实施准备、环境准备、系统切换准备、检查与支持四个步骤进行。

最后需要制定实施计划，主要包括人员配置、实施内容、实施进度、实施费用、实施结果等。为保证计划的顺利，还要制定辅助控制机制。具体计划参见表 18。

表 198 实施计划

阶段	任务	工作量(人/天)	周期(天)	结果与产出
第一阶段	选取实施对象 跨部门和跨组织选择	1.5	1.5	文档
	系统在实施对象初步实施部署和分发	5	2.5	文档

表 208 施工作计划（续）

	系统使用培训	1	1	
	解决缺陷和异常问题	8	4	文档
	完成并提交阶段一实施结果的报告	3	3	初步模型
第二阶段	实施部署剩余部门	4	2	文档
	系统使用培训	1	1	
	解决缺陷和异常问题	8	4	文档
	完成并提交阶段二实施结果的报告	2	2	结果报告
总计		33.5	21	

5.3.实施组织及培训

由于本项目属于公司内部自主开发，所以需要软件部门和运维部门作为实施人员，同时需要人力资源部门提供协助工作。对于培训工作，会由开发部门撰写使用手册，并进行培训。具体工作流程参见表 19。

表 219 实施人员准备

人员	职务职称	工作职责
甲	地区主管	负责项目整体协调，负责部门之间沟通协作
乙	项目总监	负责项目资源的协调配置；
丙	项目经理	负责项目的实施和组织
丁	技术经理	负责技术选型和系统设计

5.4.实施环境

本系统实施环境主要包括网络平台设备：一台外网接入路由器(H3C MSR 20-20)、一台 H3C 防火墙(H3C_SecPath F100-A)、一台核心交换机（D-link 3426）、一台 DMZ 区接入交换机(D-link 1252)及三台连接中心用户终端的接入交换机（D-link 1252），用于接入路由、防火墙、中心交换机与各接入交换机设备的按照方案的组网调试；用户终端使用集团办公配备的工作笔记本或台式机，无需特殊准备 PC。

内网通过网络服务提供商所提供光纤接入 Internet。根据运维部门沟通要求，系统上线时需对防火墙及接入交换机进行相应的配置，使其所连接服务器能够安全可靠地访问 Internet，且保证与集团内部其他系统之间的访问正常，同时将客户端软件安装包上传至 FTP 服务器并保证能够正常被访问，方便用户下载和安装客户端。

实施环境硬件如表 20 所示，软件配置表如表 21 所示。

表 20 硬件环境配置表

序号	系统名称	型号	详细配置	数量
服务器后台系统				
1	数据库服务器	DELL 2950	CPU 2*INTEL XEON 5355 4*4G FBD 内存 5*256G 15000 转 SAS	1
2	邮件服务器	DELL 2950	CPU 1*INTEL XEON 5355 2*4G FBD 内存 3*128G 15000 转 SAS	1
3	网站数据库服务器	DELL 2950	CPU 2*INTEL XEON 5355 2*4G FBD 内存 3*128G 15000 转 SAS	1
4	业务办公系统应用服务器	DELL 2950	CPU 1*INTEL XEON 5355 2*4G FBD 内存 3*256G 15000 转 SAS	1
网络系统				
1	D-Link 路由	DES-3550	内网核心交换机 *4 个多模光纤	1
2	D-Link 路由	DES-1252	接入交换机 *1 多模光纤	8
3	D-Link 路由	DGS-3426	外网核心交换机 *4 个多模光纤	1
4	华为 3COM	RT-MSR2020-AC-H3+LIC-S	H3C MSR 20-20 256M 内存	1

表 20 硬件环境配置表（续）

5	华为 3COM	NS-SecPath F100-A-AC	防火墙 H3C SecPath F100-A	1
其他终端设备				
1	DELL 台式机	DELL 台式机	4GB DDR3 内存 256GB SATA 硬盘 WIN XP SP2/WIN 7 系统	20
2	DELL 笔记本		4G DDR3 内存 500G SATA 硬盘 WIN 7 系统	20
2	MACBOOK 笔记本			20

表 21 软件环境配置表

序号	设备名称	型号	配置说明	数量
1	应用服务器系统	WinServer 2012	WinServer 2012	1
2	数据库服务器系统	RedHat Linux AD 5.4	RedHat Linux AD5.4	1
3	应用中间件	NC 中间件		2
4	数据库软件系统	Oracle10 企业版	Oracle10 企业版	1
5	邮件系统			1
7	防病毒软件			1

5.5.数据准备

服务器上线前，后端研发同事已经提前根据公司人力资源部门和运维部门提供的信息将基础组织架构导入至数据库，并将表结构初始化完毕。该信息数据主要包括部门信息详情、人员信息详情、日常业务数据、快捷入口地址等。

5.6.系统上线

系统在内部测试完成以后，即关闭测试服务器并切换至正式服务器。由于系统数据都是真实导入，所以不需要重复录入数据库。同时由于服务器重启，已经登陆的用户会失去连接，在重新登陆客户端即可链接到正式服务器环境。

5.7. 实施效果评估

本系统实施后，基本达到了集团的预期目标。在初期小范围试用测试中，针对系统的功能、性能、功能、容错、维护等方面都做了详细的检验，基本功能符合预期并成功切换至正式服务器，最终完成上线。软件的使用得到了同事们的认可，同时也收获了中肯的改进意见。本系统后续还会持续更新，与其他内部系统平台打通，提供更多平台快捷入口，提高办公效率。

第六章 总结与展望

本项目的需求紧紧围绕公司内部业务发展现状并进行了深度挖掘分析，为了解决现有日常办公软件通信系统的定制化高成本高、子公司数据不能共享、各系统对接困难等问题成功的设计并实施了企业自主研发的即时通讯软件系统，在截稿时产品已经成功上线，在满足了企业办公需求之余，还得到了同事们的一致好评。

本人在系统的设计与实施当中承担的主要工作内容有：

1. 收集并分析市场上常用聊天软件，结合本集团实际办公场景进行需求分析设计了软件自上而下的架构；基于 Duilib 界面库进行界面设计；使用 C/C++ 语言设计底层核心逻辑和引擎模块；
2. 设计并实现登陆模块、会话管理模块和消息管理器模块；
3. 实现了基于 C/S 架构的即时通讯软件，对接服务器接口并参与协议设计。

本系统完成了预期的建设目标，实施效果符合预期结果。在项目完成之时学习并累计了一定的界面设计和架构设计经验。目前软件还有进一步迭代开发的任务，同时页存在一定的不足需要进一步优化，主要包括以下几个方面：

1. 由于开发过程中的时间限制，界面交互设计略显单调，且开发过过程中发现 Duilib 还存在一些待改进的 bug，待后续进一步优化和完善。
2. 系统虽达到了预期，但存在部分模块的耦合度仍然较高，存在进一步分解的空间。数据缓存使用的数据结构有待进一步优化和抽象。
3. 目前仅支持普通消息类型，但通信协议的设计已经考虑了扩展性。后续需要开发和接入比如邮件提醒，会议提醒等自定义消息，需要持续进行版本迭代和开发以逐渐完善产品。
4. 项目离完美还有很长一段路要走，优化后的完全版本一定会作为一款产品与大家见面。

参考文献

- [1]. 吴海青. 基于 Webkit 内核的手机浏览器的设计与实现[D]. 北京邮电大学, 2011
- [2]. 郭东升, 田秀华. Linux 环境下基于 Socket 的网络通信[J]. 软件导刊, 2009(8)
- [3]. 高静, 段会川. JSON 数据传输效率研究[J]. 计算机工程与设计, 2011(7)
- [4]. 陈玮, 贾宗璞. 利用 JSON 降低 XML 数据冗余的研究[J]. 计算机应用与软件, 2012(9)
- [5]. 张涛, 黄强, 毛磊雅, 等. 一个基于 JSON 的对象序列化算法[J]. 计算机工程与应用, 2007(5).
- [6]. 王建军. 基于 SSL 的文件安全传输系统研究与设计[D]. 成都理工大学, 2012
- [7]. 程新党, 程强, 黄河涛. 基于 Windows 图形驱动的屏幕截图技术[N]. 南阳师范学院学报, 2005 (社会科学版)
- [8]. 张国超, 吴限, 熊芯, 等. 基于 WindML 的屏幕任意矩形区域截图技术[J]. 应用科技期刊, 2016(5)
- [9]. 张漫琪, 肖狄虎. 基于白领工作形态的办公软件设计研究[J]. 设计期刊, 2015(3)
- [10]. 胡纲. 构建企业内部通信平台[企业 QQ] [J]. 个人电脑期刊, 2003(8)
- [11]. 武萍. 即时通讯软件在企业管理中的应用——以 QQ 为例[J]. 办公室业务, 2015(2)
- [12]. 付剑锋. 基于 XML 的用户界面管理的研究与实现[D]. 上海海事大学, 2007
- [13]. 曹晓丽. 基于 DES 的加密算法[J]. 电脑知识与技术, 2011(02)
- [14]. 林辉. 基于 XML 的用户界面建模研究与实现[D]. 浙江工业大学, 2011
- [15]. 杭欣静, 蒋泽军, 王丽芳. 基于 XML 的用户界面动态生成框架的设计与实现 . 科学技术与工程期刊, 2009(5)
- [16]. 于淼. 基于 DirectUI 的界面库的设计与实现[D]. 北京交通大学, 2015
- [17]. 陈志泊, 林健. 基于 DirectUI 可扩展应用程序架构的设计与实现[J]. 计算机科学期刊, 2012(11)
- [18]. 谭成基. DirectUI 框架开发与应用[D]. 华南理工大学, 2013
- [19]. 吴湫丁. 冯运仿. 邱阳. 等. DirectUI 技术的研究与界面库设计[J]. 福建电脑, 2014(7)
- [20]. 刘俊翔. 基于 TCP/IP 的局域网即时通信软件设计[D]. 吉林大学, 2012
- [21]. 龙恒. 企业版即时通讯系统的设计与实现[D]. 吉林大学, 2015
- [22]. 杜松波. 企业即时通讯系统服务器的设计与实现[D]. 电子科技大学, 2004
- [23]. 许燕. 基于钉钉的移动办公微应用的设计与实现[J]. 软件期刊, 2017(4)
- [24]. 孙彦钊, 王彪. 利用钉钉构建高校辅导员工作网络管理平台的可行性探析[J]. 科教导刊, 2016(3)
- [25]. 舒文琼. Facebook 巨资收购 WhatsApp 微信全球拓展受阻[J]. 通信世界, 2014(3)

致谢

衷心感谢我的导师XXXX对我论文完成的辛勤培养和教导！在课题的立项初期，X老师耐心的帮助我进行选题和思路分析，并在后续论文的撰写过程中不断帮助我解决了边缘化的隐藏问题，保证了论文的逻辑严谨性和质量。这些帮助和教诲令我受益匪浅，不仅在生活中树立了正确的解决问题的习惯，而且对自己的工作有着持续的积极影响，在此，我向X老师致以最衷心的感谢！

另外，要感谢我的企业导师XX，他总是帮助我和鼓励我多进行思考，白盒化处理问题，在我撰写论文时耐心为我指导，并帮助我掌握新的技能。还要感谢默默支持我的家人，是他们为分担了更多的家庭事务，没有他们的帮助，我是无法顺利的完成项目的开发和论文的书写工作。

最后，诚挚地感谢学院评阅人和参加答辩的各位专家、教授，感谢你们的包容和理解以及给予我提出的宝贵意见。