# 项目背景

提到即时通讯，人们自然会想到微信、QQ等，但对于企业管理者而言，微信、QQ等个人即时通讯工具却存在很大隐患。一方面，员工利用这些工具做一些和工作无关的事情，从而严重影响工作效率。另一方面，QQ、微信等主要面向个人用户，非企业用户，其安全性、可管理性等方面也存在很大隐患，比如聊天过程中重要消息或者附件的泄露，这些都会给企业带来损失。此外，随着企业的发展，很多时候，企业希望即时通讯平台不是一个独立的平台，而是一个能与其他不同平台紧密相连、相互融合的即时通讯解决方案。

# 功能架构



# 产品名称

Enterprise integrated instant messaging，企业集成即时通信

# 功能概述

## 聊天平台

* + 1. 登录功能

1. 所有的平台的访问都必须通过登录后才可使用。所以系统需要提供登录功能。
2. 对于聊天端的用户，在第一次登录后系统提供登录信息记录功能，当用户下次打开时，系统提供自动登录功能。
3. 系统对于用户的登录提供审计功能。审计内容包括：1）移动端，需要记录设备号、登录账号、登录时间。2）对于PC端，需要记录到PC的IP地址、登录账号以及登录时间
   * 1. 聊天功能
4. 支持企业员工之间任何时间、任何地点进行即时交流，真正实现企业内部协同工作。
5. 支持消息格式：文字，表情，图片，附件，同时先乣支持支持离线传输。
6. 支持聊天消息提醒功能，当有新的消息到来时，能够及时通知用户。
7. 支持单聊和群聊方式。
8. 支持聊天内容的回执功能，即时通讯消息的发送，当消息发送到对端用户后，提供已发送消息回执机制，确保即时通讯消息可靠发送到对方。
9. 如果是新用户聊天，用户可以直接在客户端的通讯录中点击聊用户登录成功后可以进入个人中心对自己的信息进行管理，比如修改昵称，或者修改个人头像，同时也允许修改个人登录密码。天用户进入聊天室。
10. 在聊天室中，用户可以根据实际需要添加其他用户进入聊天室
    * 1. 在线状态

系统能够实现：

1. 文件传送每个用户可以自己管理自己的状态，比如在线，繁忙，离线等状态。
2. 每个用户可以看到聊天对方的状态
   * 1. 全文检索
3. 每个用户可以在聊天界面输入关键字进行全文检索。
4. 全文检索对象可以是通讯录用户、聊天内容等。
   * 1. 个人中心
5. 用户登录成功后可以进入个人中心对自己的信息进行管理，比如修改昵称，或者修改个人头像，同时也允许修改个人登录密码以及系统的登录登出。

## 管理平台

* + 1. 组织管理

1. 便于在IM系统中通讯用户的管理，系统需要提供组织架构管理的功能，该功能主要实现按照树状结构在系统中实现企业的组织层级架构管理。
2. 组织架构管理功能包括组织的新增、修改和删除。
3. 该功能提供WEB端的管理功能。
   * 1. 员工管理
4. 员工管理主要实现对于使用本系统进行通讯的用户进行人员信息的管理，主要信息包括：姓名，联系电话，岗位，邮件地址，所在组织架构等。
5. 员工管理主要功能包括：新增，修改，删除和离职。
6. 该功能仅仅提供PC端实现。
   * 1. 通讯录管理
7. 为方便用户在即时通讯时能够快速、方便地查找聊天用户，系统提供以通讯录的形式展现系统内的所有员工。
8. 通讯录管理提供快速查找、电话以及发起聊天的功能。
9. 该功能仅仅在PC或者移动端的聊天工具中实现。
   * 1. 账号管理
10. 账号管理主要实现对于聊天用户登录账号的管理。
11. 账号管理主要提供账号的开设、启用和禁用，密码的初始化。
12. 该功能仅仅在PC端实现
    * 1. 密码管理
13. 密码管理主要实现系统用户可以根据需要对自己的访问密码进行设定。
14. 该功能在各个端实现。
    * 1. 权限管理

权限管理这块目前可以做的简单些，就分为管理人员和普通人员，其中普通人员仅仅可以进行聊天，管理人员除了具有普通人员的聊天功能外，还可以进行一些管理性工作，比如组织架构管理，人员管理，账号管理等。

* + 1. 统计报表

# 非功能需求

## 安全性

对于关注企业信息安全的企业即时通讯系统，采用SSL/TLS 128位安全传输[加密技术](https://baike.baidu.com/item/%E5%8A%A0%E5%AF%86%E6%8A%80%E6%9C%AF)，保障数据安全性和完整性。 通过实名制，集中认证、集中管理的安全管理系统保障企业聊天数据的安全性。

## 多平台

系统需要支持多终端之间的通讯，包括移动端和PC端，移动端目前支持IOS和Android.

## 扩展性

可扩展性是软件设计的重要的原则之一，它以添加新功能或修改完善现有功能来考虑软件的未来成长。可扩展性是软件拓展系统的能力。

系统采用成熟的框架开发接口服务和后台管理，前端APP可采用Native和HTML5代码混合实现，整体采用分层设计。支持开闭原则设计思想，便于系统的灵活配置和部署；支持插件技术, 便于系统纵向延伸和对新技术的接入。

# 运行需求

## 平台硬件需求

|  |  |
| --- | --- |
| **分类** | **配置** |
| 应用服务器 | 4核CPU, 32G内存, 1T硬盘 |
| 数据库服务器 | 4核CPU, 64G内存, 1T硬盘 |

## 平台软件需求

|  |  |  |
| --- | --- | --- |
| **分类** | **名称** | **用途** |
| 操作系统 | Linux |  |
| 中间件服务器 | Tomcat |  |
| 数据库服务器 | MySQL |  |
|  |  |  |
|  |  |  |

# 开发进度和资源安排

|  |  |  |
| --- | --- | --- |
| **任务** | **进度** | **资源** |
| 开发阶段 | 8月1日~9月15阶段日 | 1个后台工程师和2个前台工程师 |
| 测试阶段 | 9月16日~9月23日 | 1个后台工程师和2个前台工程师 |
| 验收阶段 | 9月24日~9月30日 | 1个后台工程师和2个前台工程师 |
| 交付部署 | 10月8日~10月10日 |  |

# 维护方式

在软件正式运行后，开发方提供远程技术支持于远程技术维护，特定情况下开发方需派技术人员到现场进行维护于技术支持。