**企业内部沟通交流**

**软件设计文档**

软工3A项目组

指导老师：曹春

成员：141220087 申 琳

141220057 李明亮

141220137 袁 园

141220167 左枝凯

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 备注 | 编写人 | 时间 |
| V1.0 | 初稿 | 李明亮 | 2017/7/15 |

目录

一 概述

1.1 开发背景

1.2 开发目标

1.3 参考资料

1.4 设计原则

二．需求分析

2.1 需求陈述

2.2 功能分析划分

2.3 运行环境

三．总体设计

3.1 系统建模

3.1.1 系统架构图

3.1.2  ER图（实体-联系图）

3.1.3 类图设计

3.2原型设计

3.2.1 App端界面设计

3.2.2 Web端界面设计

3.3 数据库结构设计

3.3.1 数据库E-R图

3.3.2 数据库逻辑设计

四． 实现

4.1 编码

4.1.1 代码约定

4.1.2 代码编写原则

4.2 测试要点

4.2.1 登录测试要点

4.2.2 主界面测试要点

4.3 测试结果和总结

**一 概述**

**1.1 开发背景**

移动应用（App）方兴未艾，特别是聊天沟通类的app满足了人们即时沟通交流的需求。不过这些App也将生活与工作混在一起，很多职场人士是无法处理得当。使用这类App进行工作交流的话，可能前一秒还在交流工作的事情，后一秒就和朋友闲聊，刷朋友圈去了。此外许多企业内部信息泄露或被骗的新闻不断传出，也让企业内部沟通app的市场需求越来越大。

**1.2 开发目标**

使用企业内部沟通App可开展企业内部管理，加强企业成员间更好更快地交流和互动，有效提高企业管理效率。Web端管理公司组织结构，进行公司成员管理，执行发布通知发布公共等功能；App端实现点对点通信，群组通信，任务发布，任务处理，查看公告等功能。

**1.3 参考资料**

<https://github.com/jpush/jmessage-phonegap-plugin>

《软件工程案例教程》···毕硕本 卢桂香 编著 北京大学出版社

《javascript高级程序设计》···刘韬 编著 人民邮电出版社

《软件工程教材》 ···张海藩 倪宁 编著 人民邮电出版社

**1.4 设计原则**

支持通过二次开发动态增加功能模块。

基于非关系型数据库进行数据存储

后台部署基于linux关系系统

移动应用支持Android和iOS平台，Web端支持主流浏览器

**二． 需求分析**

**2.1 需求陈述**

APP端：用户登录/注册、点对点聊天、群聊、部长可开启群聊, 发布任务，处理任务，查看公告等。

Web端：后台管理公司人员增减，领导任命，公司组织结构，发布公告等。

**2.2 功能分析划分**

1．App端

（1）点对点通信

公司用户可以在公司成员列表中选取任意一人，进行即时通信，支持发送文字与发送图片功能。

（2）群组通信

每个部门可以进行群聊，部长具有创建群聊的权限，而普通部员没有创建群聊的权限。如果该部门没有群，则部员不可进行群聊；若部长选择创建群聊，则部门成员可以在部门群里进行群聊。

（3）任务管理

用户可依策略创建、启动和结束任务。通过任务管理，可以让团队成员轻松了解谁该怎什么时间内完成什么任务，并通过App进行任务处理（任务后台管理使用Activiti等流程引擎支撑）。

（4）任务处理

用户可查看自己被分配的任务，可处理待办任务。任务处理包括填写表单，提交，转派等操作。

（5）查看公告

用户可以在App客户端查看管理员推送的公告，后台管理员针对组织结构树中某一层次的用户发布通知公告等，在App端可以解析为markdown格式。

2.Web端

（1）组织结构管理

定义企业树状分层结构，将子部门作为父部门的叶节点，父部门作为子部门的父亲结点，由管理员在浏览器中操作，对企业的组织结构进行管理，并以以上所提到的树的结构可视化企业的部门。

（2）用户管理

创建，修改，删除用户，可禁用或启用某用户，可将用户放置入组织结构中，由管理员在浏览器中操作，可指定某部门中一个人作为部长。

（3）任务管理

任务后台管理使用Activiti等流程引擎支撑，在web端时刻监视每个任务的状态。

（4）信息发布

可针对组织结构树中某一层次用户发布通知、公告等信息，由管理员在浏览器中操作，信息以Markdown格式定义

**2.3 运行环境**

1.硬件环境

App端支持移动设备（手机、平板电脑）；web端支持移动设备与PC；

2．软件环境

App端支持IOS/Android操作系统（支持版本IOS和Android版本X以上）Web端支持主流浏览器；后台部署支持linux操作系统；数据存储使用mongoDB非关系型数据库。

**三． 总体设计**

**3.1 系统建模**

**3.1.1 架构设计图**

1.视图层（View）

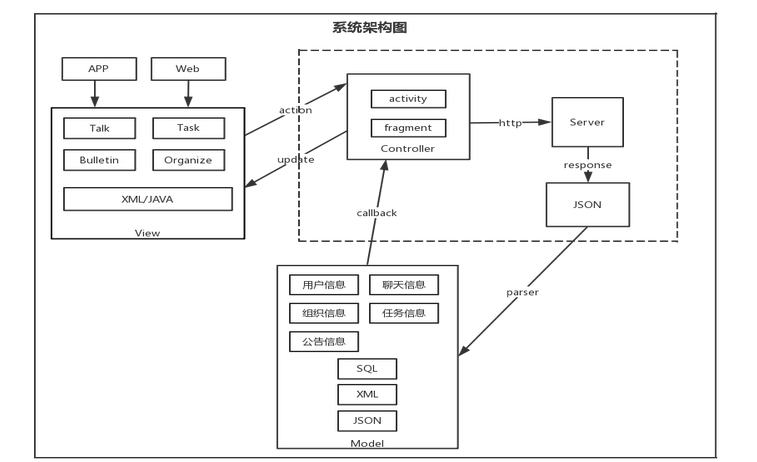
是提供给用户的操作界面，是程序的外壳。界面就是各种UI组件（XML布局或者Java自定义控件对象）。

2. 数据层（Model）

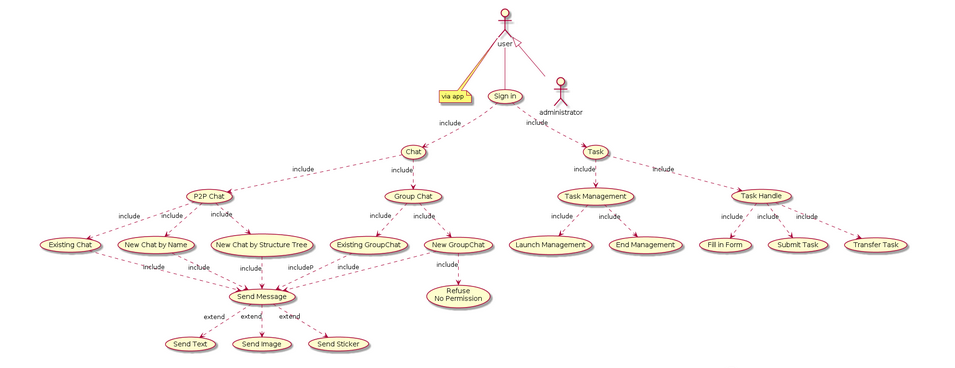
是程序需要操作的数据或信息（系统中的业务逻辑部分）。通常是数据库SQLite、网络请求的JSON、本地XML或者Java对象数据。

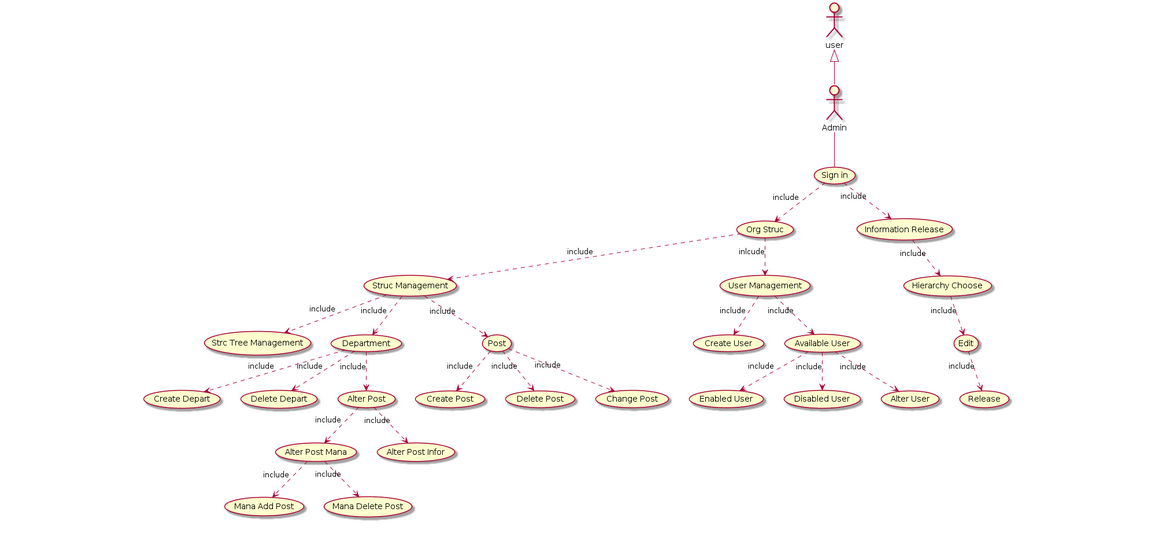
3. 控制层（controller）

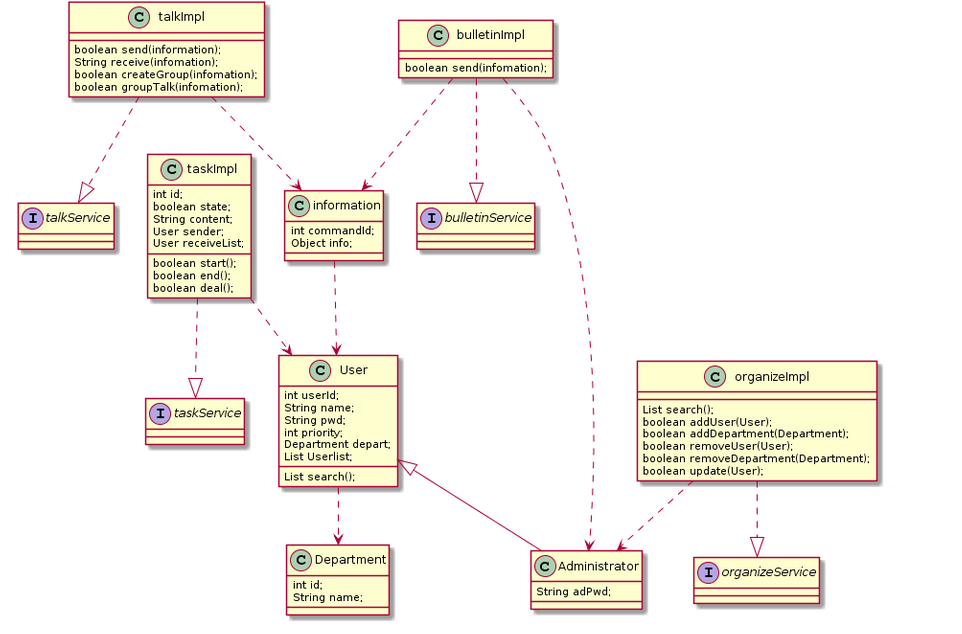
负责根据用户从“视图层”输入的指令，选取“数据层”中的数据，然后对其进行相应的操作，产生最终的结果。控制器是与应用程序相关联的动作集合，负责处理待响应的请求。通过界面响应用户输入，通过模型层处理数据，最后返回结果给界面。



**3.1.2  ER图（实体-联系图）**





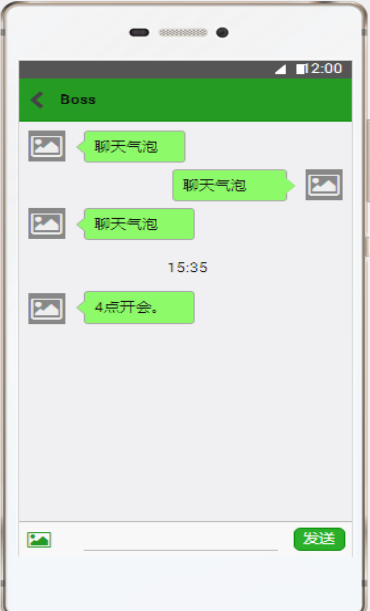
**3.1.3 类图设计**

**3.2原型设计**

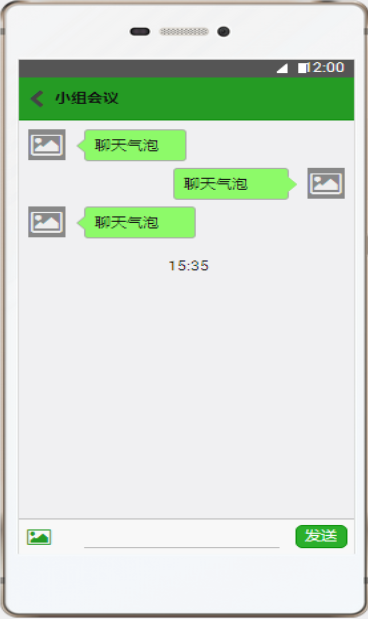
**3.2.1 App端原型设计**

1)登录，进入APP，初始为聊天列表



**2）**搜索用户ID或者直接点击选择用户进行聊天

**3）**发起群聊，直接进行部门选择，不用单独选择用户，进入群聊



**4）**任务创建，查看当前已有任务状态



**5）**待处理任务查看以及处理



**6）**个人信息，可修改密码，查看公告，退出登录





**3.2.2 Web端界面设计**

**https://modao.cc/app/Zglxj9Y3kVCm4xx4vmGp8Szo3fXmcn1#screen=s1D58CF3E4D1490444814960**

**3.3 数据库结构设计**

【主要是描述】

**3.3.1 数据库E-R图**

**3.3.2 数据库逻辑设计**

**四． 实现**

**4.1 编码**

**4.1.1 代码约定**

**4.1.2 代码编写原则**

**4.2 测试要点**

**4.2.1 登录测试要点**

【描述该如何测试，数据的输入，类型】

**4.2.2 主界面测试要点**

**……**

**4.3 测试结果和总结**