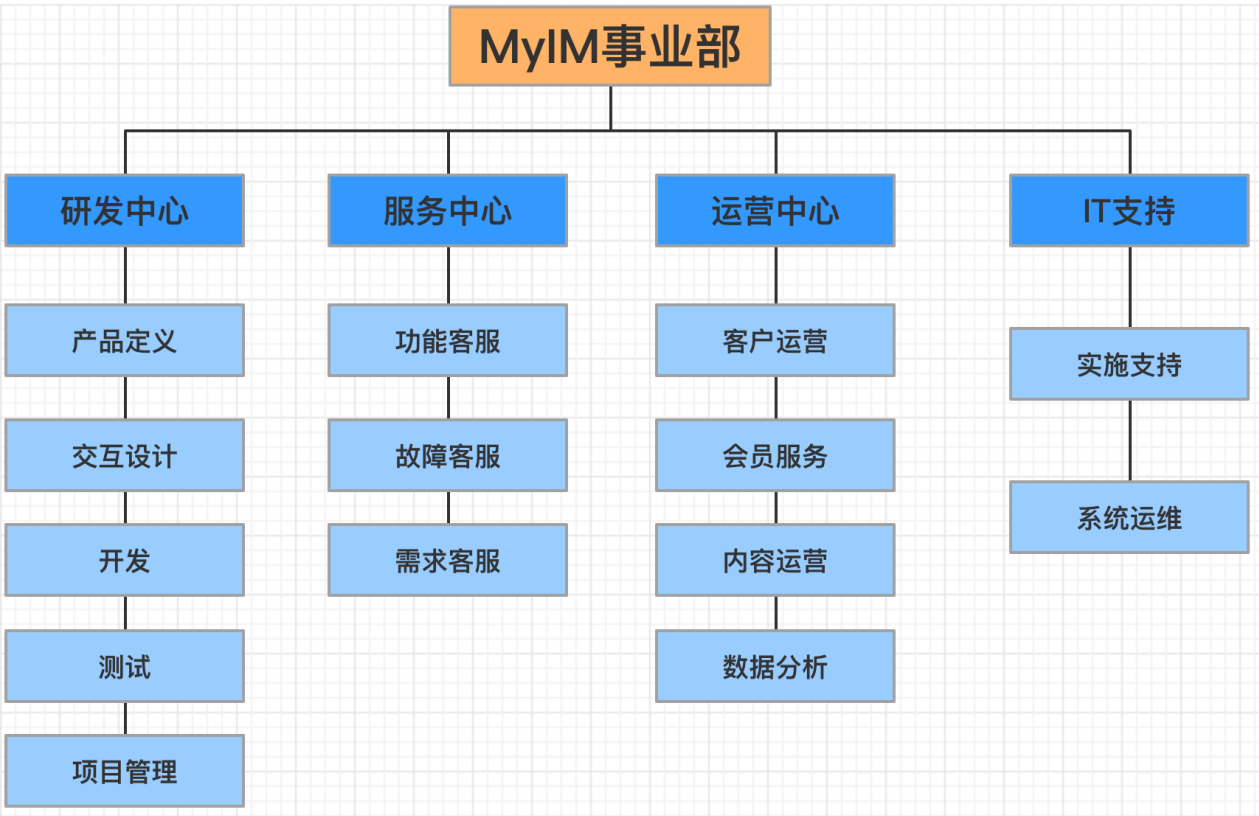


# 一、数据库应用系统规划

## 1.1 系统命名

MyIM 即时通讯系统

## 1.2 组织架构



即时通讯业务属于小巧专注的垂直领域，所以部门设计较为扁平化，主要对用户场景进行分工支持。

- 研发中心负责设计定义产品、开发测试产品以及项目管理，推动产品面世
- 服务中心负责接收用户的反馈与意见，为产品改进提供数据
- 运营中心负责产品运营，寻找盈利点，提高产品使用体验
- IT支持部门则产品上线后的后续部署运维等工作

## 1.3 业务场景

用户可以使用我们的通讯工具的APP、小程序、网页端等客户端，首先进入登录界面，当用户已经有账号时可以输入其邮箱和密码后点击登录按钮进入主页面，主页面中会展示当前用户发生的会话列表，点击任意一个会话可以进入消息的详情页面，在该页面中可以查看已经发送与接收的消息，并且可以通过底部的输入框编辑并发送新的消息。在主页面的右上角可以点击搜索与添加按钮，点击搜索按钮后进入搜索页面，输入要搜索的用户邮箱或群号可以查找到已注册的用户和群，点击添加按钮后

可以创建新的群聊。在任意界面中，点击用户或群头像可以进入详细介绍页面，用户的详细介绍页会展示该用户的ID、邮箱、昵称、性别、生日等信息，群的详细介绍页会展示该群的ID、名称、群主，群成员等信息。

## 1.4 性能指标

通讯场景下的压力主要来自于高并发场景下的网络通信的压力以及消息发达的速度与精度，软件系统本身的开销基本上可以忽略，对终端设备的性能要求很低。

- 可同时接收用户注册请求数量
- 可同时接收用户登录请求数量
- 可同时处理用户间消息发送数量
- 其他数据交换的高并发请求处理能力
- 消息发送到接收的时延
- 消息的丢失率

## 1.5 战略地位

即时通讯系统的本质是构建一个用户间沟通的平台，依托于这个平台可以拓展出许多的衍射业务

- 独立的通讯工具，通过提供高效的通讯体验来吸引用户，提供公司的用户量，并以此来转化到公司的其他业务中
- 拓展的社交平台，以聊天为中心开发社群、交友、兴趣组等社交，并提供会员服务
- 嵌入的客服工具，可以为公司的其他产品提供通讯需求的技术支持，模块化嵌入使用，提高其他产品的竞争力

## 1.6 运营分析

由于该系统的重点难点均在于计算机技术实现而不依赖于硬件设备，因此主要的支出在于软件设计开发成本、运营推广成本以及运维成本。按照目前常见的软件工程师、设计师、运营人员等薪资待遇，预估每年的人力成本支持约在50万左右，各种推广活动按照常见的广告收费约在10万左右，服务器等硬件成本每年应在1至2万，后续用户量提升会逐渐增加服务器开支，预计能承受每10万用户的访问量的机器成本应该在每年5-10万左右。

在收益方法，直接收益即为会员服务与广告收入，每10万用户按照10%的会员率估计每年的会员费收入约120万左右，广告收入则与投递商有关，每接收一个广告投递所获得的盈利约在10w左右。除此之外还有为公司其他产品提供支持而产生的隐性收入，此类收入无法计算，但是不可忽略

## 1.7 技术选型

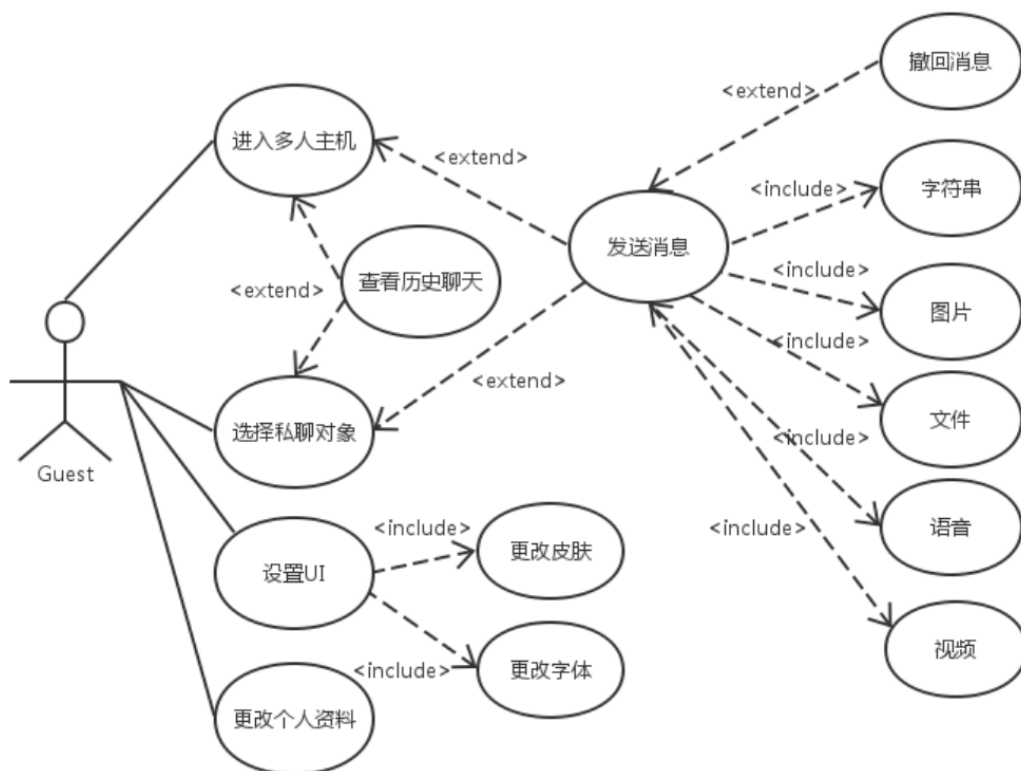
- 客户端方面使用较为流行的uniapp框架进行开发，对于初创公司而言开发成本低周期短，一次开发即可部署到手机端、小程序、网页端等，节省了开发成本的同时，又能做到多端覆盖
- 服务端方面采用GO语言进行开发，因为该系统的主要压力来自于高并发量，而Go语言在处理高并发方面有着天然的优势，并且其相比JAVA而言语言简洁，开发周期短，相比Python又具有很强的性能与稳定性，长远来看随着用户量的增长也不用担心系统无法承受。
- 在数据库方面采用MySQL存储主要的数据库。MySQL是目前业界最为流行的关系型数据库，经过了大量公司的许多项目的考验，因此在稳定性上不需担心。同时可能会结合Redis等NoSQL数据库来处理非关系型的数据，如消息内容等实时性较高的数据，获得更高的处理速度。

## 1.8 专家意见

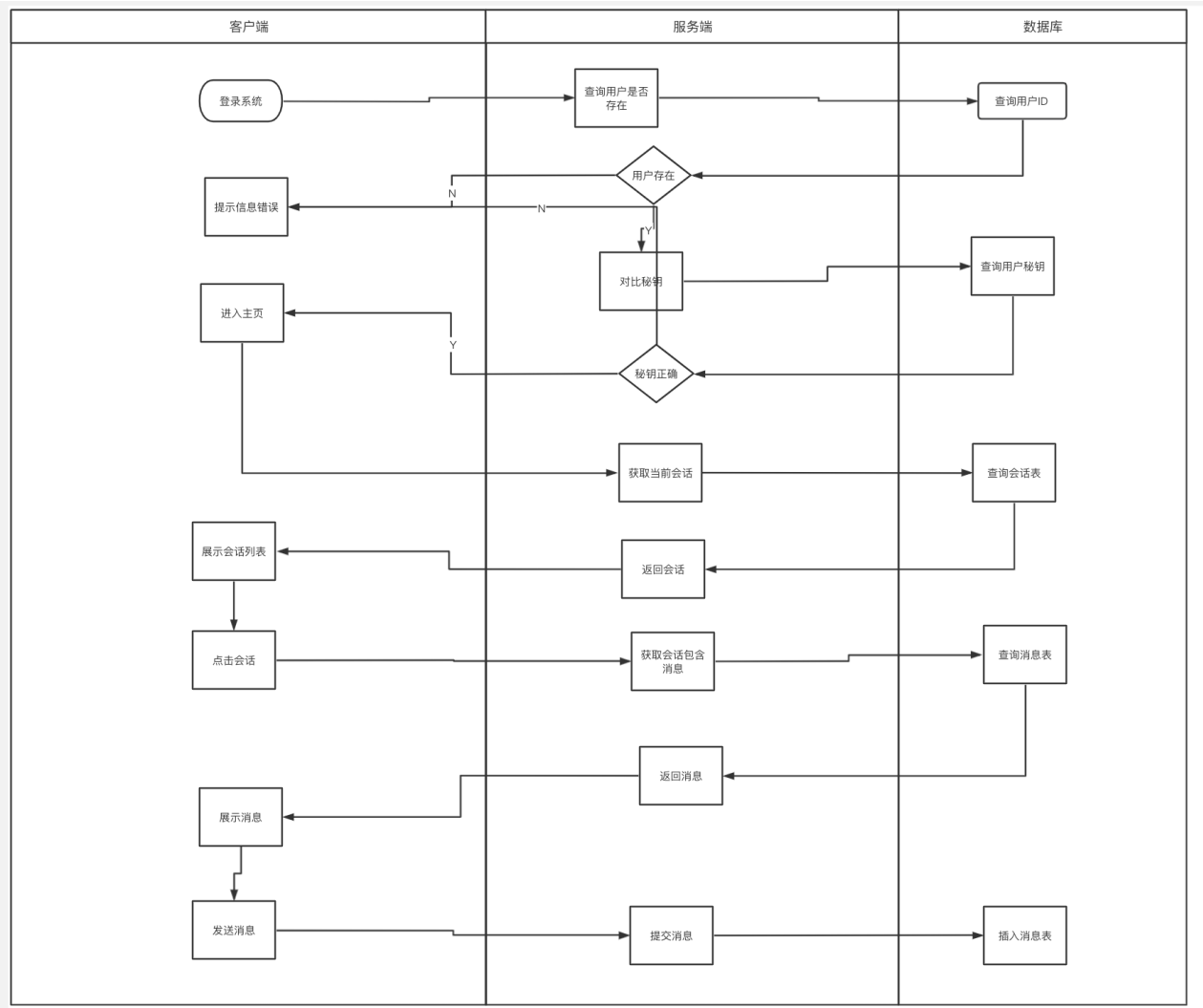
1. 刘鸿羽：认为总体上较为合理，在数据库方面也可以结合SQLite来存储本地数据，不需要将所有数据都存到服务器上。
2. 何仕超：认为使用Go语言开发是非常好的决定,并且建议直接使用Go进行原生Server开发而不要使用框架，因为Go的原生Web开发已经非常优秀了。

# 二、数据库应用系统需求分析

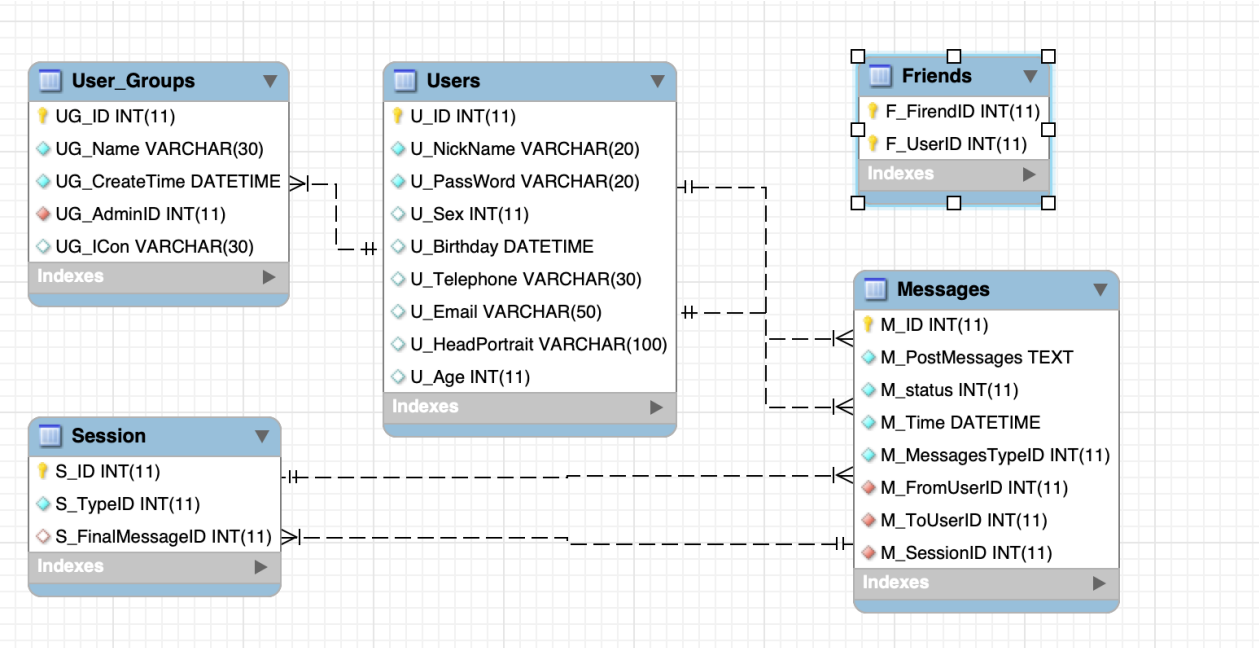
## 2.1 业务需求



2.2 业务场景

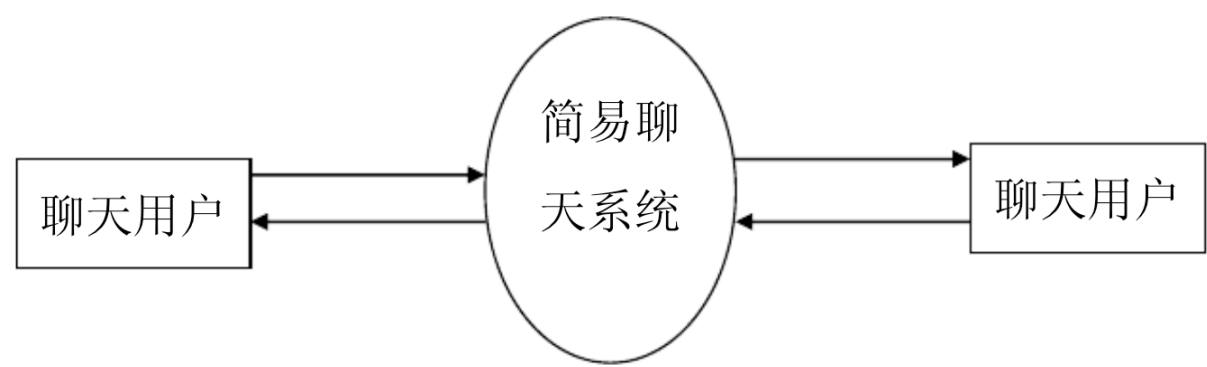


2.3 数据结构

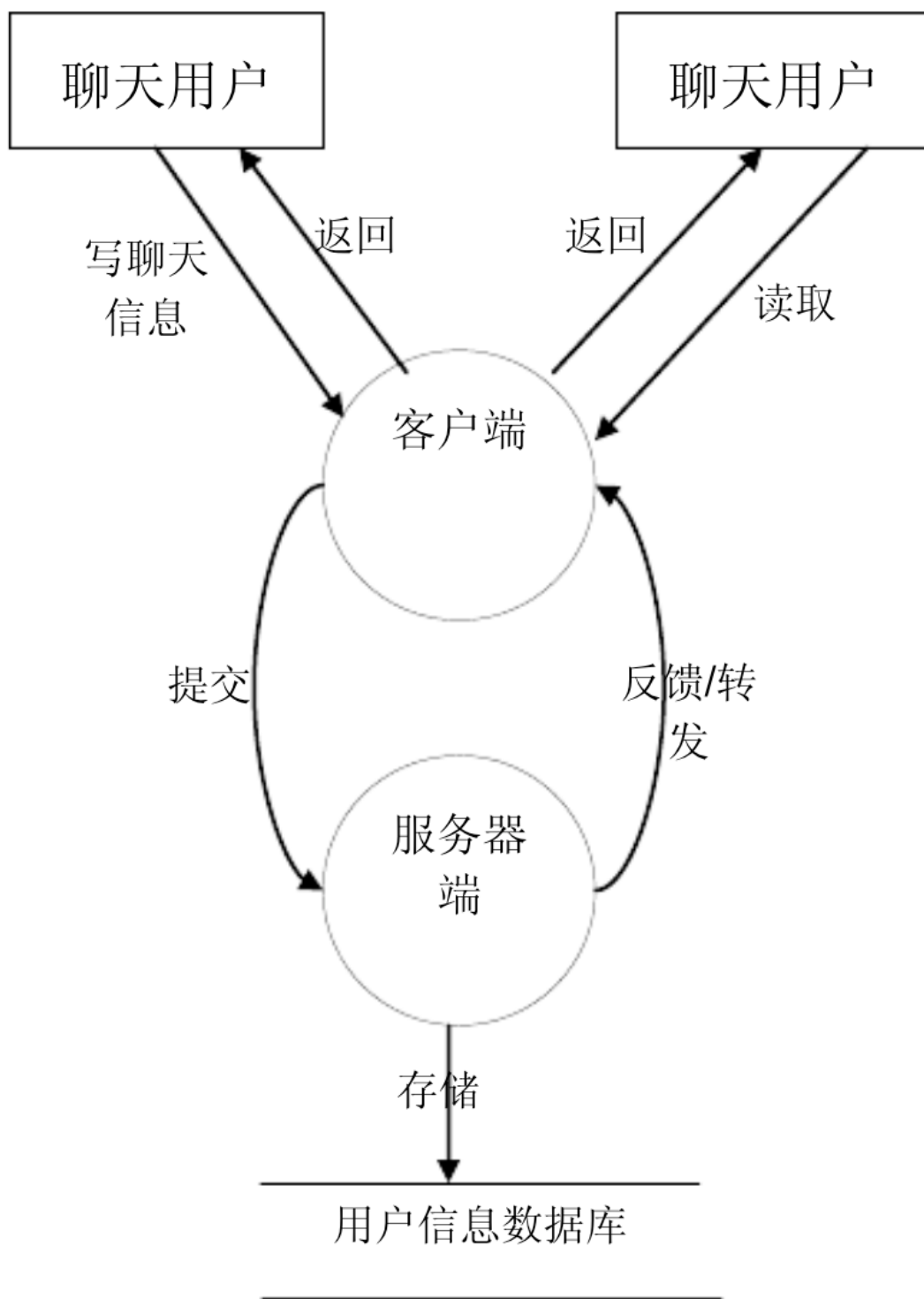


2.4 数据处理

2.4.1 顶层数据流图

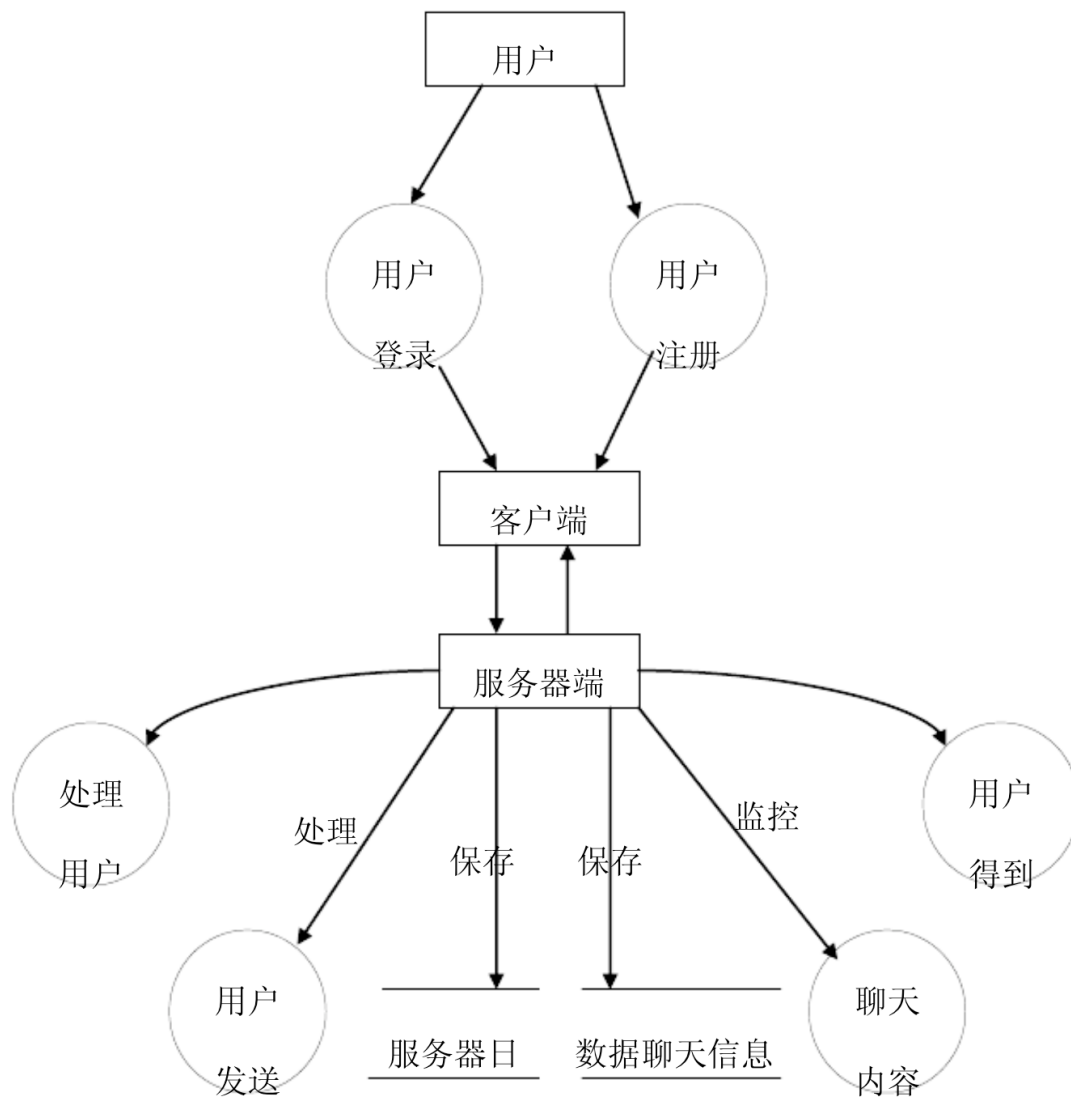


2.4.2 一层数据流图



一层数据流图

### 2.4.3 二层数据流图



## 2.5 非功能性需求

### 2.5.1 业务处理性能需求

支持多用户的高并发

### 2.5.2 安全性

避免消息内容丢失与泄露

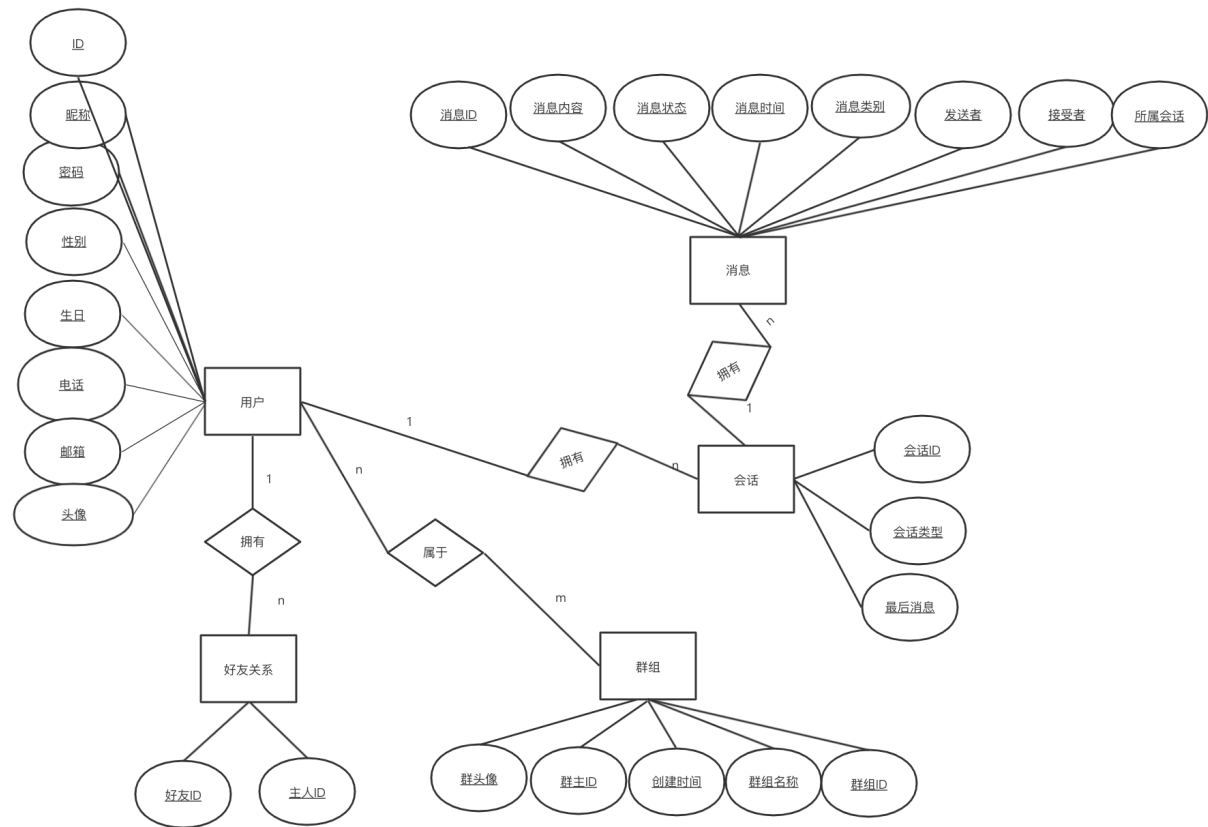
### 2.5.3 完整性

满足数据模式的完整性

# 三、数据库应用系统设计

## 3.1 实体关系模型

设计的数据库已经充分满足了需求，如下是对应的实体关系图



## 3.2 数据库规范

### 3.2.1 第一范式

数据库中的每一列都是不可分割的基本数据项

原先设计的数据表已经充分满足了第一范式

### 3.2.2 第二范式

每个实例或行必须可以被唯一的区分

原先设计的数据表已经充分满足了第二范式



### 3.2.3 第三范式

不包含已在其他表中已包含的非主关键字信息

原先设计的数据表已经充分满足了第三范式