**1 问题定义**

**1.1项目背景与意义**

（1）N(Need 需求)：随着互联网的发展，人们越来越离不开各种各样的手机应用与软件，它们渐渐融入我们的生活，但随着科技的日渐飞跃，人际交流却陷入低谷，用户更少地与他人交流，沟通能力也在日益丧失，另外人与人之间沟通的效率也在降低，用户希望可以有一个程序可以更好地与他人交流并且可以实现人机交互。

（2）A(Approach 做法)：通过制作智能聊天室的软件实现用户与机器的聊天以及用户与用户之间的交流互动。

（3）B(Benefit 好处)：聊天机器人提高了用户获取信息的有效性，深刻的改变着人们的生活方式，促进了人机的互动和交流，并且帮助用户更好地了解、获取信息，以及使交流更加人性化。

（4）C(Competitors 竞争)：市面上现在有许多类似的聊天软件，本软件可以在用户无人聊天的时候与机器人聊天，不会感到寂寞孤独。

（5）D(Delivery 推广)：在同学群和家长群中进行推广宣传。

**1.2 项目基本目标**

制作出一个类QQ的程序客户端，服务端能够显示已连接的用户，并向指定用户甚至所有用户发送信息，通过切换发送对象能够进行随时随心的私聊。客户端的聊天界面可以显示每条信息的发送时间，好友的用户名等信息。并且可以与聊天机器人随时随心进行聊天。

**1.3可行性分析**

（1）经济可行性：该聊天工具是一个小型系统，只需有一台电脑就可以开发出来，因此实际成本可以忽略不计。

（2）技术可行性：该聊天系统采用python语言开发编程。现有技术如python、socket套接字编程可以支持完成软件设计，从技术层面看，此软件是可行的。

（3）需求可行性：智能聊天室可以满足人们即使的聊天需求，帮助用户在特定的环境下获取外界信息，存在需求，从需求层面看，此软件是可行的。

（4）开发方案可行性：目前有很多信息提供了类似的聊天技术和开发方案，可以为软件开发提供很多值得借鉴的经验帮助用来更好的开发，从开发方案层面看是可行的。

**2需求分析**

2.1服务端需求分析

（1）处理用户在客户端发送给服务端的消息并精准的向客户进行转发

（2）在服务端数据库中保存用户的注册账号和注册密码。

2.2客户端需求分析

（1）注册账号，注册密码。

（2）向指定客户发送消息。

（3）实现聊天机器人与用户的信息交互。

2.3 E-R图

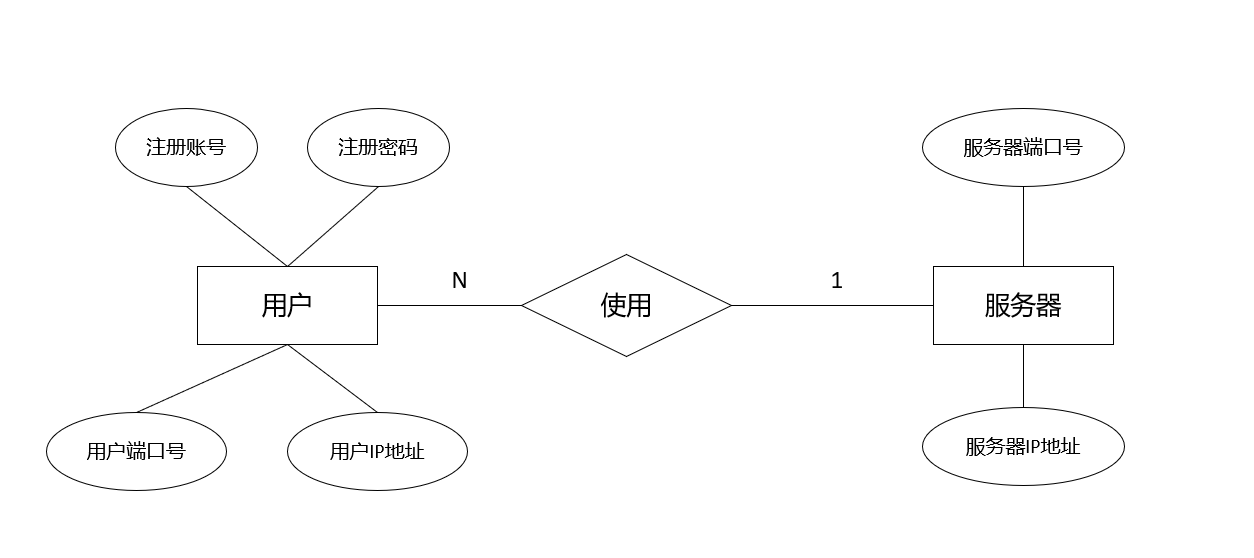


图1 E-R图

2.4数据流图

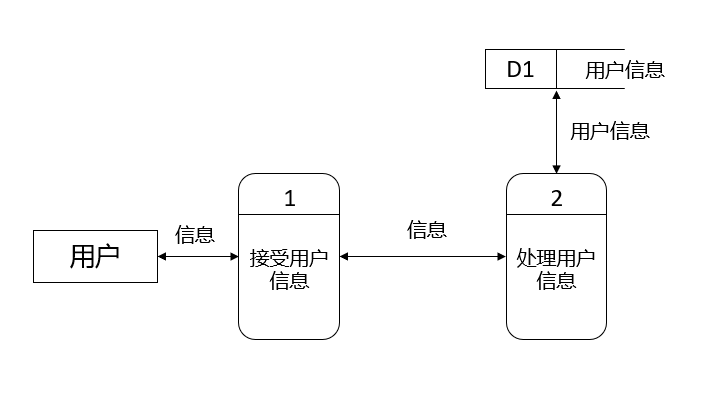


图2 数据流图

2.5用例图

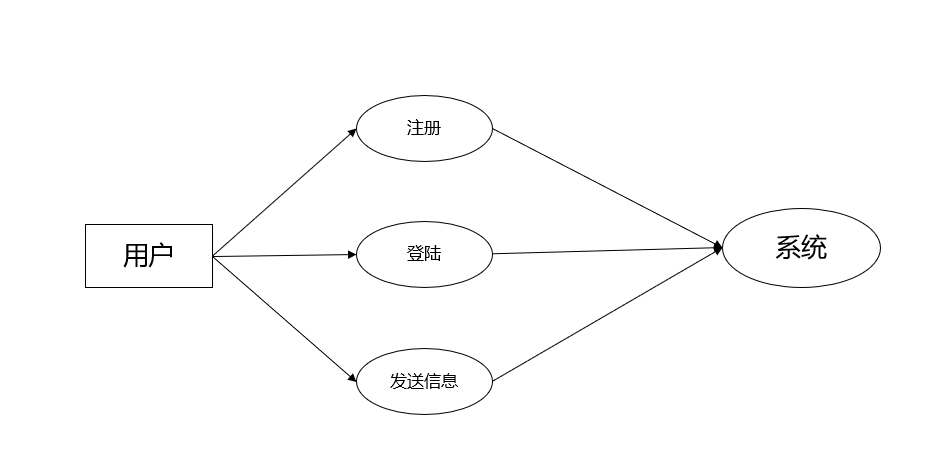


图3 用户用例图

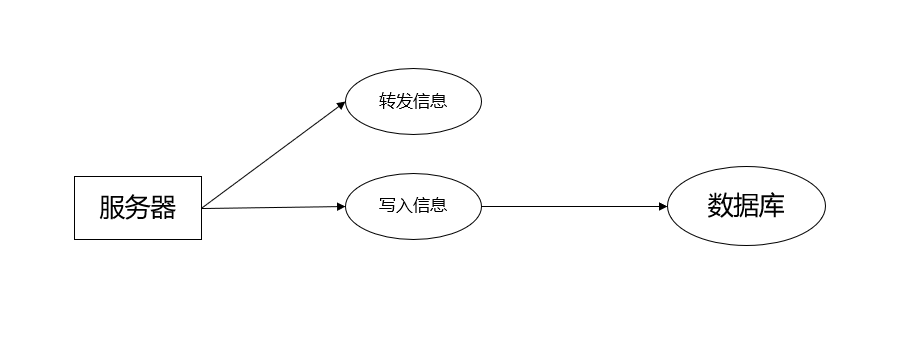


图4 服务器用例图

2.6 原型系统设计

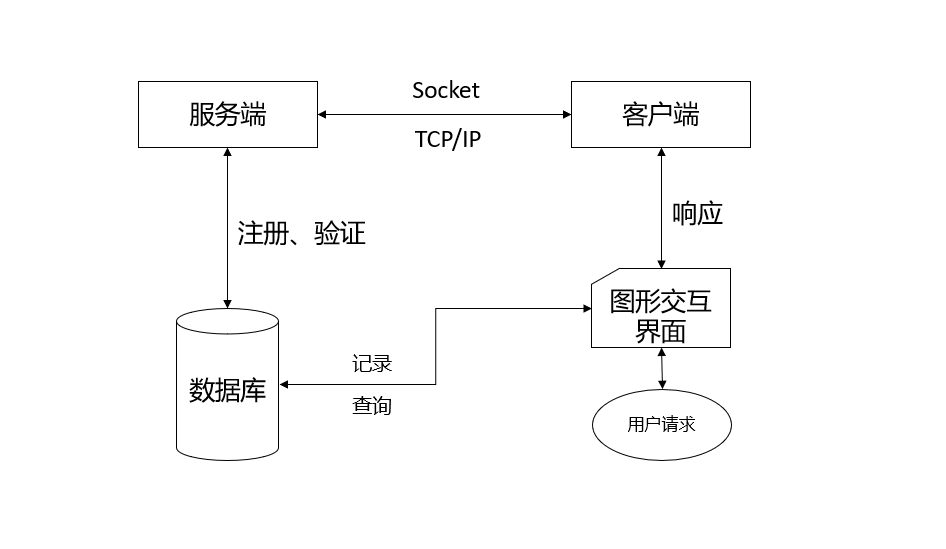


图5 原型系统设计图

概要设计和详细设计

3.1系统结构

智能聊天室是以python为内核，使用web技术封装的聊天软件，实现了智能聊天。

智能聊天室的主体系统分为两个部分，第一个是客户端（client）和服务端(server)，负责接收消息和发送消息，第二个是智能聊天机器人，负责接收用户发送的消息并通过计算返回信息。

3.1.1功能说明

智能聊天室的功能为用户可以与聊天机器人进行实时对话，主要步骤为：用户通过GUI界面登陆聊天室，若无账号可以选择注册，有账号可直接登陆，登陆后可以选择与聊天机器人进行聊天，发送消息给聊天机器人后，聊天机器人会自动识别内容然后计算返回相应信息，实现实时聊天的功能。

3.2接口设计

智能聊天室的接口主要为信息传输和聊天机器人得劲接口设计，首先用户登录账号后，向聊天机器人发送信息，此时为用户端向服务器发送信息，当服务器收到信息立即将信息转发给聊天机器人，聊天机器人经过运算后通过服务端返回信息。

主要的设计思想为：client端输入信息传输给server端，server端接收到client端的信息后将信息转发给聊天机器人，聊天机器人收到信息后运算生成信息给server端，再由server端将信息传输给client端，用户在client端收到信息后继续输入信息，实现用户与聊天机器人在智能聊天室里的信息交互。

3.3关键数据结构设计

在client和server中，设计了消息的传输机制。主要为在server中设计了两个socket，一个是用来监听的listensocket,另一个是用来进行消息传输的datasocket。

3.4关键算法设计

关键算法为server和client的消息传输算法设计。

在server中，首先实现了用户注册和登录以及用户相关信息的管理功能。其次使用多线程函数，当client发送消息给server时唤醒server，server首先设定一个IP地址和端口号，使用bind方法将主机地址和端口号进行绑定，同时创建listensocket使令服务器处于监听状态，随时接受来自于client端的消息。

在client中，用户需要进行登录，成功后能发送消息。具体的算法设计为，先创建套接字，指定IP地址和端口号，使用socket函数指明协议，同时使用connect函数与server进行连接。

Server接收到连接之后创建一个datasocket进行client与server的数据之间的消息传输。

3.5数据管理说明

用户再使用智能聊天室时，会被要求先登录账号，如果为新账号则可以使用注册功能，输入用户名和密码就可注册账号。用户的账号和密码将会一律被保存至user.json文件中，等用户下一次登录时，输入用户名和密码后后台会在user.json文件中进行匹配，若匹配成功，则用户能成功登录，若匹配不成功，用户则不能成功登录。

4.实现与测试

4.1实现环境和代码管理

实现环境为pycharm+anaconda3,代码管理为github平台。

4.2 关键函数说明

def sign\_up(client):

此函数为client中的用户注册函数

def login(client):

此函数为client中的用户号登录函数

def serverFunc(conn):

此函数为server中的多线程函数，主要作用为接受client端发来的消息，启动此函数。

4.3测试计划和测试用例

4.4结果分析

经过测试，智能聊天室能正常的进行登录、注册，实现与聊天机器人进行自动的聊天，并且聊天机器人回复速度良好，能在于用户的聊天交互过程中实现迅速和实质性的回复。

5.总结

5.1用户反馈

在经过多次试验后，整个软件的流程逐渐变得流畅，在于聊天机器人的对话也十分迅速，能实现用户再聊天室总自由与机器人聊天的功能，但是由于聊天机器人需要训练，囿于此原因，机器人的训练量可能不足，因此有时候会回复重复性的内容，但是这个问题能通过不断的训练聊天机器人得到解决。

5.2总结

在整个实验过程中，发现在进行实践之前需要提前做好系统设计和需求分析等，实验的实践过程是以系统设计为基础，需求分析为导向的，在不断的迭代过程中，产品才会有进步，通过不断的反馈和不断的修正，才能最终打磨出一个好产品。