

一、项目概述：

此为 sci 课程的第 6 项目小组的关于实现异步聊天的以数据库为支撑的聊天程序，实现了私聊与群聊的功能。并增加了查看好友、加删好友、查看私聊的未读信息、创建群聊、加入群聊等细化功能。

该文档描述的是最新版本的项目设计，旧版本的项目因为还未完善到可以实现的地步，这里就不提供旧版本的描述了。

二、设计思路与想法：

设计 类实现对网民、群聊的基本信息的储存；类图：

```
class Netizen;

class Group
{
public:
    Group(std::string name); //构造函数
    bool addNetizen(std::string groupname, Netizen *n); //添加某个特定群聊的群成员
    bool Judge(std::string groupname, Netizen *n); //判断是否存在该群聊,存在则添加该用户为群成员
    bool hasGroup(std::string groupname); //判断是否有这个群聊名
    void checkName(); //输出群聊的群成员名
    std::string getName(); //getter groupname

private:
    std::string m_name; //群聊名
    std::vector<Netizen *> _netizens; //用于储存一个群聊的群成员
};

class Group;

class Netizen
{
public:
    Netizen(std::string name, std::string id);

    void createGroup(pqxx::connection &A); //创建群聊
    void joinGroup(pqxx::connection &A); // 加入群聊
    void checkGroups(pqxx::connection &A); // 查看群聊
    void chooseGroupToChat(pqxx::connection &A); //选择群聊去聊天
    void watchNotRead(pqxx::connection &A, std::string tablename); //查看未读信息
    void chat(pqxx::connection &A, std::string groupname); //聊天
    void insertInformation(pqxx::connection &A, std::string tablename, std::string information);

    bool sameName(std::string l_name); //在netizens中找到某个人,判断提供的人名是否在netizens中
    void output(); //输出用户信息
    void outputFriends(); //输出用户好友信息,从类中操作

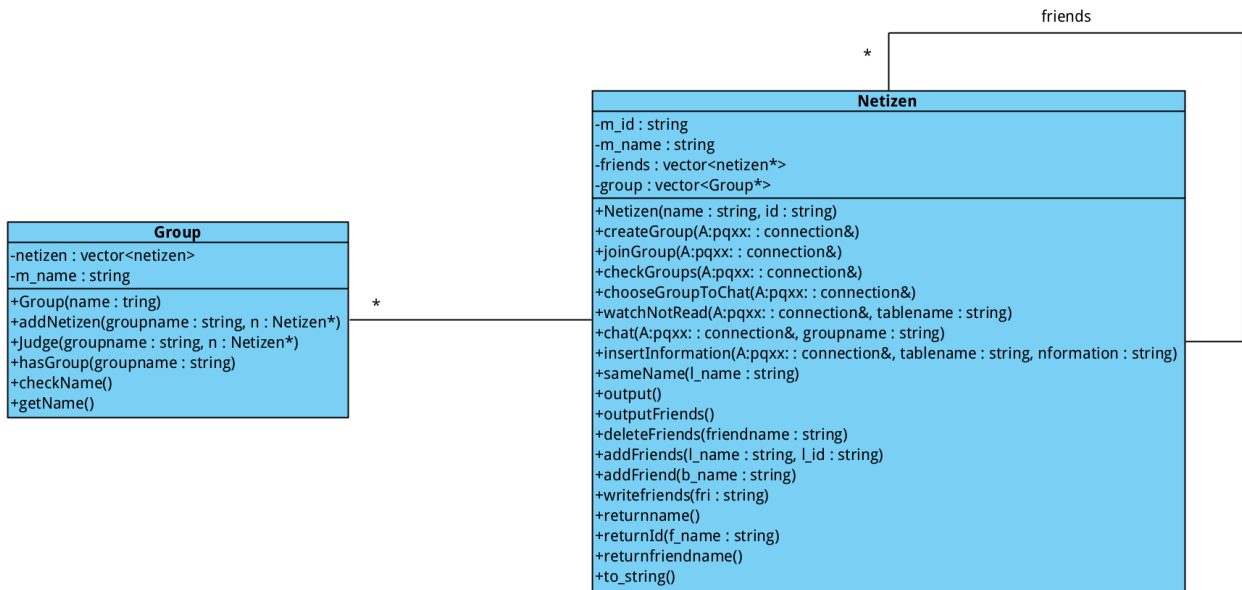
    void deleteFriends(std::string friendname); //删好友,从类中操作

    void addFriends(std::string l_name, std::string l_id); //添加朋友(从文件读入时)
    void addFriend(std::string b_name); //加好友,在netizens数组中操作
    void writefriends(std::string fri); //将数据库中的好友信息写入类,用于dataRead函数

    std::string returnname(); //getter m_name
    std::string returnId(std::string f_name); //如果名字存在,返回id,在netizens中操作
    std::string returnfriendname(); //返回netizens中好友的名字

    std::string to_string(); //将用户的id和name连成字符串
private:
    std::string m_id;
    std::string m_name;
    std::vector<Netizen> _friends; //用于存储用户的朋友信息
    std::vector<Group *> _groups; //用于存储用户加入过的群聊名
};

void read(pqxx::connection &c);
```



设计 数据库来储存聊天消息和基类基本信息等数据，防止数据变化时数据的不同步；表的设计：

用户数据表：

Table "public.mickey"									
Column	Type	Collation	Nullable	Default	Storage	Compression	Stats target	Description	
info	character varying(50000)				extended				
sender	character varying(50)				extended				
isonline	character(3)				extended				
isnew	character(3)				extended				

Access method: heap

info 用于存放聊天信息 messages

sender 用于存放发送消息的名字

isonline 用于存放 mickey 用户是否在线的 YES/NO

isnew 用于表示该消息是否为新消息的标志 YES/NO

data 数据表：

Table "public.data"									
Column	Type	Collation	Nullable	Default	Storage	Compression	Stats target	Description	
id	character varying(30)		not null		extended				
name	character varying(50)		not null		extended				
friends	character varying(50000)				extended				

Access method: heap

id 用于存放用户的 id 信息

name 用于存放用户的 name 信息

friends 用于存放用户的朋友信息，存储的是朋友的名字

groups 数据表：

Table "public.groups"									
Column	Type	Collation	Nullable	Default	Storage	Compression	Stats target	Description	
name	character varying(300)		not null		extended				

Access method: heap

name 用于存放群聊的名字

群聊数据表：

Table "public.first1"									
Column	Type	Collation	Nullable	Default	Storage	Compression	Stats target	Description	
name	character varying(100)		not null		extended				
information	character varying(50000)				extended				
time	character varying(50)				extended				
flag	character varying(10)				extended				

Access method: heap

name 用于存放发消息的人的 name

information 用于存放发消息的人的 message

time 用于存放发消息时的 time

flag 用于表示该消息是否为新消息的标志 new/previous

这里我们先向数据库中安放了已有的信息，从已有的文件（data.dat）中读取（read(c)函数）数据放到类中，并在下一步调用 dataInsert 函数来将类中的数据存入数据库中的 data 表里。之后再运行程序时我们注释了 read()函数，增添了 dataRead 函数来读取 data 表中的信息到类里面，实现面向对象编程。当然在程序的结尾，即选择“保存退出”时，会调用 dataInsert 函数来将类中的信息存入数据库中，以同步数据。

三、具体代码实现：

程序运行的流程简述：

首先提示“登陆选 1; 作为新用户注册选 2:”因为已经录入初始信息，所以测试时，选 1;

之后显示主菜单：

```
//主菜单
void menuMain()
{
    cout << "-----" << endl;
    cout << "- Welcome to the chat room: " << endl;
    cout << "- 1. 查看我的好友 " << '\n'
    << "- 2. 加好友 " << '\n'
    << "- 3. 删好友 " << '\n'
    << "- 4. 查看私聊的未读消息 " << '\n'
    << "- 5. 私聊 " << '\n'
    << "- 6. 群聊 " << '\n'
    << "- 7. 保存退出 " << endl;
    cout << "-----" << endl;
}
```

并提示“Please select:”，基本功能不再赘述，如果选择 5,则提示“请输入你想发送消息的人的名字:”输入名字正确后就会提示“可以开始发消息了! (输入 over 结束聊天)”，就可以开始聊天了，注意，这里的聊天是异步的。如果选择 6,则显示群聊菜单：

```
//群聊菜单
void menuGroups()
{
    cout << "-----" << endl;
    cout << "- Welcome to the group chat room: " << endl;
    cout << "- 1. 创建群聊 " << '\n'
    << "- 2. 加入群聊 " << '\n'
    << "- 3. 选择群聊进行聊天 " << '\n'
    << "- 4. 退出 " << endl;
    cout << "-----" << endl;
}
```

并提示 "Please select: "接下来会有提示词，根据提示词进行操作即可。

程序结束运行请选择 7：保存退出。（注意：否则该程序不能同步数据库信息）

程序代码的详细解释：

首先进入 main.cpp：

```
std::vector<Group> groups;
std::vector<Netizen> netizens;

int main()
{
    selectFunction();
    return 0;
}
```

主要功能是设置全局变量与引用 selcetFunction 函数

其次程序跳转至 chat.cpp：进入 selectFunction 函数,执行打开数据库操作（chat.cpp 包含#include <pqxx/pqxx>头文件，运行该程序时请确保配置环境的一致），之后 dataRead 读入数据库信息，显示菜单，while 和 switch case 实现来调用各选择需要使用的函数。

如果选择 1：调用 outputFriend 函数输出。

如果选择 2：提示"请输入你想添加的好友的名字："，输入已存在的用户后调用 addFriend 函数。

如果选择 3：提示"请输入你想删除的好友的名字："，输入已存在的好友后调用 deleteFriend 函数并且显示当前已更改的好友列表。

如果选择 4：提示"以下是你未读的消息："，调用 ShowNewMessage 显示所有自己未读的消息，并提示"你想要选择其他功能吗？（y/n）："，如果选 y，那么调用 menuMain 函数显示主菜单让用户选择功能，如果选 n，那么调用 dataInsert 把更改的信息同步到数据库。

如果选择 5：提示"请输入你想发送消息的人的名字："输入已存在的用户名字，然后会先调用 ShowMessage 显示与该用户的历史记录消息，之后提示"可以开始发消息了！（输入 over 结束聊天）"，调用 Send 函数发消息。

如果选择 6：调用 selectGroupsToChat 函数进入群聊，首先设置 Netizen *n 指针，用来存储正在执行该程序的用户，以便类信息的存储和类成员函数的调用。然后显示菜单，如果选择 1：调用 n→createGroup(A)，（A 是数据库参数），如果选择 2：调用 n->joinGroup(A),如果选择 3：调用 n->chooseGroupToChat(A)，如果选择 4：退出群聊

如果选择 7：调用 setNo 函数设置该用户的数据表标识该用户的下线，然后调用 dataInsert 更新 data 和 groups 数据表。

重点模块的解释：

私聊：主要在 Send 函数中实现发消息，首先设置一个字符串 isNew 来判断该消息是否是新消息的标签，并设置布尔类型的 judge 局部变量来存储 isOnline 函数的结果，如果 judge=false,设置 isNew="YES"，表示接收方 receiver 不在线，那么 message 对 receiver 来说就是未读的新消息，发送消

息的时候就将该 message 记录在 receiver 用户数据表中并打上新消息的标记。当 receiver 登陆时查看未读消息，程序就根据该标记来输出内容。

之后设置 4 个用于数据库的执行语句字符串：

```
string flag = "insert into " + receiver + "(info) values('New messages:');";
```

//插入值到receiver 表的info 列, 值为New message

```
string currentTime = "insert into " + receiver + "(info,sender,isNew) values('' + TimeAdd()  
+ ',' + sender + ',' + 'YES');";
```

//插入值到receiver 表的info,sender,isNew 列, 设置未读标签与时间显示(TimeAdd 时间显示是对 receiver 不在线的专属, info 列存储发送消息的时间, sender 列表示发送方为 sender, 未读说明 isNew 一定是 YES

```
string Minesql = "insert into " + sender + "(info,sender,isNew) values('' + message  
+ ',' + 'You','NO');";
```

//插入值到sender 表的info,sender,isNew 列, info 列存储message, sender 列值设置为 You, 发送方 sender 一定在线, 所以是已读消息 isNew 列值为 NO

```
string Othersql = "insert into " + receiver + "(info,sender,isNew) values('' + message + ',' +  
+ sender + ',' + 'isNew + ');";
```

//插入值到receiver 表的info,sender,isNew 列, info 列存储消息 message, sender 列存储为给他发消息的人的名字, 即'sender', isNew 列存储局部变量isNew 的值

群聊：（为方便描述，称呼当前执行该程序的用户为 netizen）在创建群聊 Netizen::createGroup 函数中，首先会创建一个 groups 数据表（如果数据库中没有 groups 数据表的话）来储存群聊名称。之后会加入 groupname 到 groups 数据表中。最后会创建聊天表用来存储聊天信息，该聊天表的名称就为 groupname。

当要求用户输入要创建的群聊名称，执行 groups.push_back(groupname)语句来将 groupname 保存在全局数组 groups 中；

再调用 Group::Judge 函数判断该 groupname 是否存在 groups 数组中,如果存在就将该 netizen 的地址放入 Group 类的 _netizens 数据成员中，并返回 true；

重新回到 Netizen::createGroup 函数, if(true) 则将 groupname 地址存入该 netizen 的 _groups 数据成员中，这样 netizen 加入的群聊中就有了 groupname,而 groupname 拥有的网民就有了 netizen.实现了类的二元关联。（代码如下）：

```
groups.push_back(groupname);
```

//将群聊添加到netizen 对象的_groups 中

```
for (auto &g : groups) {
```

```
    if (g.Judge(groupname, this)) //判断群聊名是否存在, 存在就加入_groups
```

```
        Group *ptr = new Group(g);
```

```

        _groups.push_back(ptr);
    }

```

//判断是否存在这个群聊,如果存在就返回true 并添加该群成员

```

bool Group::Judge(string groupname, Netizen *n)
{
    for (auto &g : groups) {
        if (g.hasGroup(groupname)) {
            _netizens.push_back(n);
            return true;
        }
    }
    return false;
}

```

在网民加入群聊 Netizen::joinGroup 函数中，首先调用 checkGroups 函数来显示已经创建好的群聊列表，提示"请输入你想加入的群聊："之后输入 groupname，执行以下语句添加信息后 return 退出函数，netizen 和 group 的二元关联与上述代码同理：

```

for (auto &g : groups) {
    if (g.addNetizen(groupname, this)) {
        _groups.push_back(&g);
        return;
    }
}

```

//添加某个特定群聊的群成员

```

bool Group::addNetizen(string groupname, Netizen *n)
{
    if (m_name == groupname) {
        _netizens.emplace_back(n);
        return true;
    }
    return false;
}

```

在聊天 Netizen::chat 函数中，首先调用 watchNotRead 查看群聊的未读消息（与私聊同理 flag='new'），之后输入聊天消息 over 结束聊天，并调用 insertInformation 插入信息到当前 group 数据表中，插入的信息设置 flag='new'（因为是异步聊天，对当前 netizen 是 previous 消息，但对其他未登陆的 netizen 来说都是 new 消息）

注意：Netizen 和 Group 类在群聊功能实现时，数据库中没有存储一个群聊中有多少个 netizen，一个网民中有多少个 group，即群聊是不设限的，只要选择 3 调用 chooseGroupToChat 选择 groups 数组中已经存在的群聊就可以在这个群聊中聊天。即数据库中不会保存两类信息，但是程序运行中的两类信息是保存了的，只不过程序关闭就销毁了。

四、配置环境与测试：

配置环境：

操作系统：plasma6;

编译环境：qt: cmake_minimum_required(VERSION 3.30)

target_compile_features(chat PRIVATE cxx_std_23)

target_include_directories(chat PRIVATE /usr/include)

target_link_libraries(chat PRIVATE /usr/lib/libpqxx.so)

数据库：postgresql 配置：

安装：sudo pacman -S postgresql

初始化：sudo -u postgres initdb --locale en_US.UTF-8 -E UTF8 -D '/var/lib/postgres/data'

启动数据库：systemctl start postgresql

创建：sudo su postgres

createuser --interactive ChatManager

createdb chat

连接数据库：psql -U ChatManager -d chat

关闭数据库：systemctl stop postgresql.service

c++ 与 postgresql 数据库连接：pacman -S libpqxx

yay -S libpqxx

测试说明与截图：

程序运行时请在终端配置数据库！

开始测试：systemctl start postgresql

./chat


```

root ~ > chatProgram6Group > build > C_C_2025_01_04-Debug > ./chat
Opened database successfully:chat
登陆选 1; 作为新用户注册选 2: 1
请输入您的名字和ID号!
您的名字是:Mickey
您的ID号是:ntz1
-----
- Welcome to the chat room:
- 1.查看我的好友
- 2.加好友
- 3.删好友
- 4.查看私聊的未读消息
- 5.私聊
- 6.群聊
- 7.保存退出
-----
Please select: 1
Mickey的朋友有:
Donald
Goofy
Dewey

Please select: 2
请输入你想添加的好友的名字: Louie
添加朋友成功!
Please select: 3
请输入你想删除的好友的名字: Donald
删除成功, 下列是您的好友列表:
Mickey的朋友有:
Goofy
Dewey
Louie

Please select: 4
以下是你未读的消息:

你想要选择其他功能吗? (y/n): y
-----
- Welcome to the chat room:
- 1.查看我的好友
- 2.加好友
- 3.删好友
- 4.查看私聊的未读消息
- 5.私聊
- 6.群聊
- 7.保存退出
-----
Please select: █

```

图 1 (左) 图二 (右)

```

你想要选择其他功能吗? (y/n): y
-----
- Welcome to the chat room:
- 1.查看我的好友
- 2.加好友
- 3.删好友
- 4.查看私聊的未读消息
- 5.私聊
- 6.群聊
- 7.保存退出
-----
Please select: 1
Mickey的朋友有:
Goofy
Dewey
Louie

Please select: 4
以下是你未读的消息:

```

你想要选择其他功能吗? (y/n): n

```

root ~ > chatProgram6Group > build > C_C_2025_01_14-Debug >

```

当提示“你想选择其他功能吗?”时, 选择 y 会显示菜单, 用户可以继续选择功能 (如图 1); 选择 n, 那么程序直接退出。(如图 2)

```

您的名字是:Mickey
您的ID号是:ntz1
-----
- Welcome to the chat room:
- 1.查看我的好友
- 2.加好友
- 3.删好友
- 4.查看私聊的未读消息
- 5.私聊
- 6.群聊
- 7.保存退出
-----
Please select: 4
以下是你未读的消息:

你想要选择其他功能吗? (y/n): y
-----
- Welcome to the chat room:
- 1.查看我的好友
- 2.加好友
- 3.删好友
- 4.查看私聊的未读消息
- 5.私聊
- 6.群聊
- 7.保存退出
-----
Please select: 5
请输入你想发送消息的人的名字: Donald
now: -2025.1.15.0:12:39:
可以开始发消息了! (输入over结束聊天)
Hello!
Nice to meet you!
I am Mickey!
over
Please select: 7

```

```

root ~ > chatProgram6Group > build > C_C_2025_01_14-Debug >

```

当 Mickey 登陆, 选择 5 向 Donald 发起私聊。(如图 3)

当 Donald 登陆, 选择 1 查看好友。选择 4 显示出 Mickey 先前向 Donald 发送的消息, 即 Donald 未读的消息, 之后 Donald 选择 5 私聊 Mickey, 输入 over 结束后再选择 4 时显示的未读消息就为空了, 即已读。(如图 4)

```

请输入您的名字和ID号!
您的名字是:Donald
您的ID号是:ntz2
-----
- Welcome to the chat room:
- 1.查看我的好友
- 2.加好友
- 3.删好友
- 4.查看私聊的未读消息
- 5.私聊
- 6.群聊
- 7.保存退出
-----
Please select: 1
Donald的朋友有:
Mickey
Minnie
Goofy
Louie

Please select: 4
以下是你未读的消息:
(-2025.1.15.0:12:44,Mickey)
(Hello!,Mickey)
(-2025.1.15.0:12:52,Mickey)
("Nice to meet you!",Mickey)
(-2025.1.15.0:12:57,Mickey)
("I am Mickey!",Mickey)

你想要选择其他功能吗? (y/n): y
-----
- Welcome to the chat room:
- 1.查看我的好友
- 2.加好友
- 3.删好友
- 4.查看私聊的未读消息
- 5.私聊
- 6.群聊
- 7.保存退出
-----
Please select: 5
请输入你想发送消息的人的名字: Mickey
now: -2025.1.15.0:14:12:
(-2025.1.15.0:12:44,Mickey)
(Hello!,Mickey)
(-2025.1.15.0:12:52,Mickey)
("Nice to meet you!",Mickey)
(-2025.1.15.0:12:57,Mickey)
("I am Mickey!",Mickey)
可以开始发消息了! (输入over结束聊天)
Hello!
I can see you!
Nice to meet you too!
over
Please select: 4
以下是你未读的消息:

```

```

你想要选择其他功能吗? (y/n): n

```

```

root ~ > chatProgram6Group > build > C_C_2025_01_14-Debug >

```

图 3 (左) 图 4 (右)


```
您的名字是:Mickey
您的ID号是:ntz1
-----
- Welcome to the chat room:
- 1.查看我的好友
- 2.加好友
- 3.删好友
- 4.查看私聊的未读消息
- 5.私聊
- 6.群聊
- 7.保存退出
-----
Please select: 6
-----
- Welcome to the group chat room:
- 1.创建群聊
- 2.加入群聊
- 3.选择群聊进行聊天
- 4.退出
-----
Please select: 1
请输入要创建的群聊名称:third
创建群聊成功,已退出创建群聊!
Please select: 2
Chat groups list:
first
second
third
请输入你想加入的群聊:
first
加入群聊成功!
Please select: 3
以下是您已经加入过的群聊:
third
first
请输入要加入聊天的群聊名称:
first
可以开始聊天了!(输入over结束聊天)
Hello!
I am Mickey!
Have somebody there?
over
依然在这个群聊聊天吗?(y/n)
n
已退出聊天!
Please select: 4
已成功退出!
-----
- Welcome to the chat room:
- 1.查看我的好友
- 2.加好友
- 3.删好友
- 4.查看私聊的未读消息
- 5.私聊
- 6.群聊
- 7.保存退出
-----
Please select: 7
```

root@ubuntu:~/chatProgram6Group> build> C C 2025 01 14-Debug

图 5 (左) 图 6 (右)

当 Mickey 选择 6 时进入群聊菜单, 选择 1 创建群聊 third; 选择 2 加入群聊 first, (事先已经创建了一些群聊, 因为当用户创建群聊时他就已经加入该群聊了, 所以需要先创建一些群聊供用户选择) 之后选择 3 显示 Mickey 加入过的群聊, 提示选择要加入聊天的群聊名称, Mickey 选择 first, 然后就开始在 first 群中开始聊天。(如图 5)

当 Donald 登陆选择群聊菜单, 选择加入 first 群聊, 就会显示 first 群聊的消息, 当然 Donald 也可发送信息 (如图 6)

```
您的名字是:Donald
您的ID号是:ntz2
-----
- Welcome to the chat room:
- 1.查看我的好友
- 2.加好友
- 3.删好友
- 4.查看私聊的未读消息
- 5.私聊
- 6.群聊
- 7.保存退出
-----
Please select: 6
-----
- Welcome to the group chat room:
- 1.创建群聊
- 2.加入群聊
- 3.选择群聊进行聊天
- 4.退出
-----
Please select: 2
Chat groups list:
first
second
third
请输入你想加入的群聊:
first
加入群聊成功!
Please select: 3
以下是您已经加入过的群聊:
first
请输入要加入聊天的群聊名称:
first
可以开始聊天了!(输入over结束聊天)
(Mickey>Hello!)
(Mickey,I)
(Mickey,am)
(Mickey,Mickey!)
(Mickey,Have)
(Mickey,somebody)
(Mickey,there?)
Oh,
I am here.
Hello,Mickey!
over
依然在这个群聊聊天吗?(y/n)
n
已退出聊天!
Please select: 4
已成功退出!
-----
- Welcome to the chat room:
- 1.查看我的好友
- 2.加好友
- 3.删好友
- 4.查看私聊的未读消息
- 5.私聊
- 6.群聊
- 7.保存退出
```

数据表的内容演示：如下图

chat=# \dt

List of relations			
Schema	Name	Type	Owner
public	daisy	table	ChatManager
public	data	table	ChatManager
public	dewey	table	ChatManager
public	donald	table	ChatManager
public	first	table	ChatManager
public	goofy	table	ChatManager
public	groups	table	ChatManager
public	huey	table	ChatManager
public	louie	table	ChatManager
public	mickey	table	ChatManager
public	minnie	table	ChatManager
public	pluto	table	ChatManager
public	second	table	ChatManager
public	third	table	ChatManager

(14 rows)

chat=# SELECT * FROM data;

id	name	friends
Mickey	ntz1	Donald Goofy Dewey
Donald	ntz2	Mickey Minnie Goofy Louie
Minnie	ntz3	
Goofy	ntz4	Mickey Donald
Pluto	ntz5	
Daisy	ntz6	
Huey	ntz7	
Dewey	ntz8	
Louie	ntz9	

(9 rows)

chat=# SELECT * FROM groups;

name
first
second
third

(3 rows)

chat=# SELECT * FROM mickey;

info	sender	isonline	isnew
Hello!	You	NO	NO
Nice to meet you!	You	NO	NO
I am Mickey!	You	NO	NO
New messages:		NO	
-2025.1.15.0:14:22	Donald	NO	YES
Hello!	Donald	NO	YES
New messages:		NO	
-2025.1.15.0:14:37	Donald	NO	YES
I can see you!	Donald	NO	YES
New messages:		NO	
-2025.1.15.0:14:50	Donald	NO	YES
Nice to meet you too!	Donald	NO	YES

(12 rows)

chat=# SELECT * FROM donald;

info	sender	isonline	isnew
New messages:		NO	NO
-2025.1.15.0:12:44	Mickey	NO	NO
Hello!	Mickey	NO	NO
New messages:		NO	NO
-2025.1.15.0:12:52	Mickey	NO	NO
Nice to meet you!	Mickey	NO	NO
New messages:		NO	NO
-2025.1.15.0:12:57	Mickey	NO	NO
I am Mickey!	Mickey	NO	NO
Hello!	You	NO	NO
I can see you!	You	NO	NO
Nice to meet you too!	You	NO	NO

(12 rows)

chat=# SELECT * FROM first;

name	information	time	flag
Mickey	Hello!	-2025.01.15 00:24:31	previous
Mickey	I	-2025.01.15 00:24:37	previous
Mickey	am	-2025.01.15 00:24:38	previous
Mickey	Mickey!	-2025.01.15 00:24:38	previous
Mickey	Have	-2025.01.15 00:25:07	previous
Mickey	somebody	-2025.01.15 00:25:07	previous
Mickey	there?	-2025.01.15 00:25:07	previous
Donald	Oh,	-2025.01.15 00:26:18	new
Donald	I	-2025.01.15 00:26:24	new
Donald	am	-2025.01.15 00:26:24	new
Donald	here.	-2025.01.15 00:26:24	new
Donald	Hello,Mickey!	-2025.01.15 00:26:30	new

(12 rows)

图 7 (左) 图 8 (右)

五、任务分工与心得体会：

首先感谢我们第六项目小组的努力付出：

郭瑞桦：用户相关信息查询，私聊用文件实现部分

王婷：群聊功能的实现

陈敏：对之前基于文件的私聊进行优化改为使用数据库的版本与版本整合

其次分享心得体会：

实际上每一个小的功能都面临着思考者和设计者极大的脑力风暴，稍微一个细节出现差错就会导致整个程序的崩溃，这不仅要求我们对技术与语言的熟练运用，更考验我们的情绪是否冷静，能够稳定的分析问题，共同解决问题。以及数据的整合，对于不同变量的定义怎么快速准确的修改和整合，怎样让运行不会出现差错也很考验代码水平。其次，团队交流与协作分工同样重要，如何在最省力省时的情况下对代码的设计做出调整，如何融合所有人的创意并分工实现代码，如何进行思维的碰撞提升自己的编程水平和合作能力都是我们面临的，也是正在慢慢感受的问题。

总之，这次的项目让我们收获颇丰，也对项目开发有了初步的理解，学习到了很多课外知识，也理解老师的良苦用心，祝愿我们大家都拥有更加坚定的意志，走向理想的道路！

编至 2025.1.14