

TEAM 13 2016312860 KimSeonji
2017311456 KimSoyeong

CONTENTS

**Basic
Techniques**

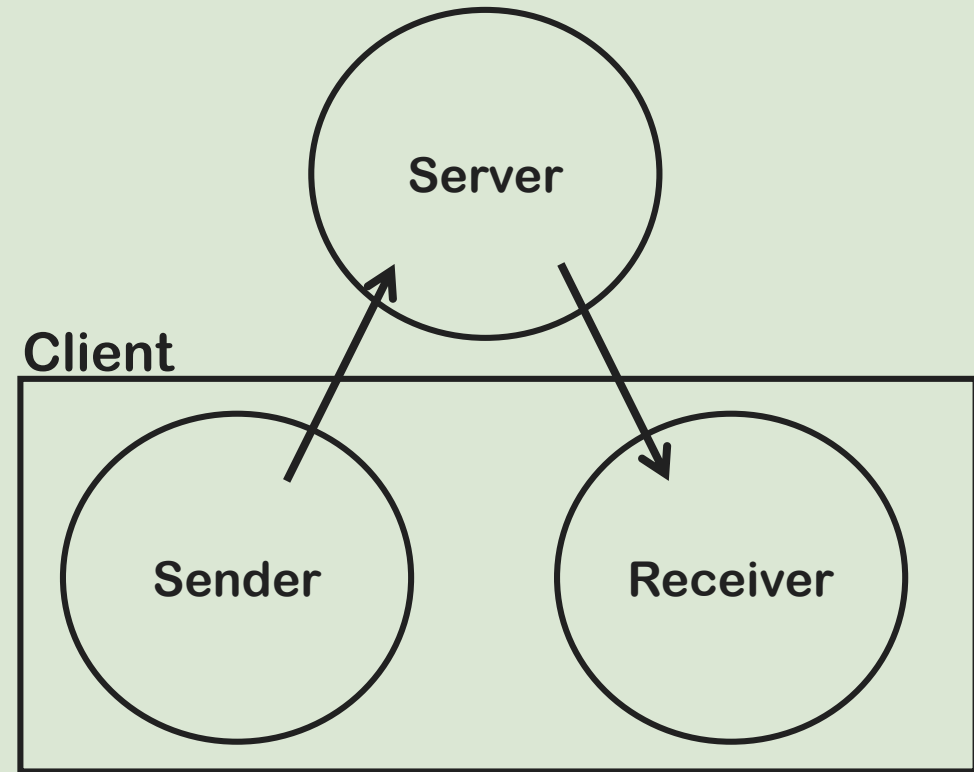
**Application
Execution**

Application

FlowChart

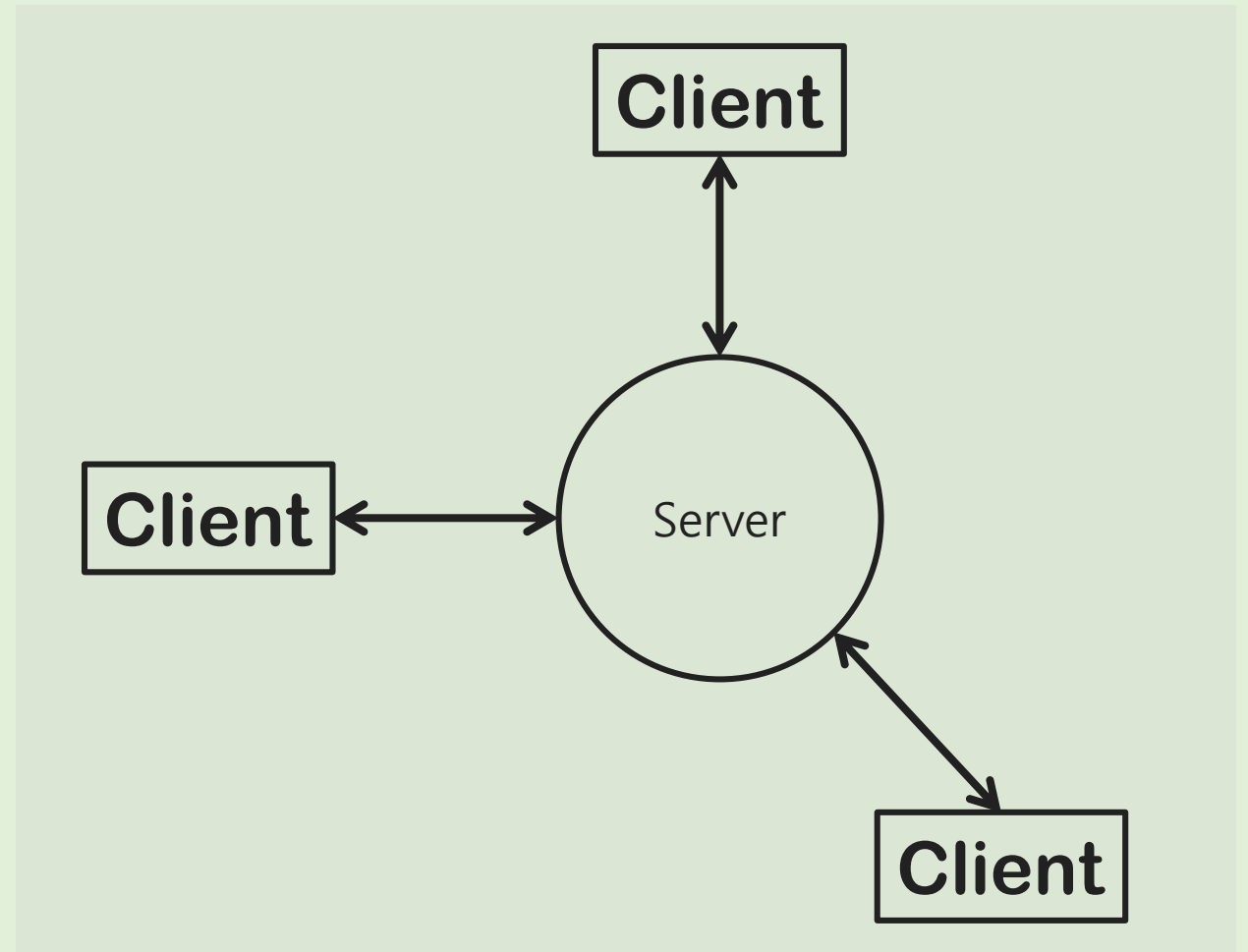
Basic Techniques

- **Socket Programming**
- Multithreading
- HashMap
- Command parsing
- Chat State



Basic Techniques

- Socket Programming
- **Multithreading**
- HashMap
- Command parsing
- Chat State



Basic Techniques

- Socket Programming
- Multithreading
- **HashMap**
- Command parsing
- Chat State

globalMap	HashMap <String, HashMap<String, ServerRecThread>>
clientMap	HashMap<String, ServerRecThread>
history	HashMap<String, String>
joinuser	<u>HashMap</u> <String, String>
loginuser	<u>HashMap</u> <String, Integer>
cl	HashMap<String, String>

Basic Techniques

- Socket Programming
- Multithreading
- HashMap
- Command parsing
- Chat State

Command : #IN name(of room)

Inner command : Req_enterRoom|name|#IN name(of room)

Split("|");

Req_enterRoom

name

#IN name

name	#IN name(of room)
------	-------------------

Split(" ");

#IN	name(of room)
-----	---------------

Basic Techniques

- Socket Programming
- Multithreading
- HashMap
- Command parsing
- **Chat State**

0	before login
1	after login = HOME
3	chatting
4	1:1 answering
5	send file
6	1:1 checking
7	Going home
10	Report bad user
100	Whisper

Basic Application(Requirements)	Additional Application
<ol style="list-style-type: none">1. 1 : M (Send Letter)2. N : M (Make new Room)3. Sign up4. Save Conversation5. Communication List6. File transfer	<ol style="list-style-type: none">1. page [HOME]2. Show Login/Join user list3. Logout and Withdraw4. 1 : 1 Chat5. Report Bad users6. Whisper

Application Execution

Sign up

- JoinUser HashMap

- ◆ Checking ID

- ◆ Checking Password

Client1 : apple

```
WELCOME SundiSodi
Are you our member? [YES/NO]
no
[##]Please be a member of <SundiSodi>
▶ ID :
apple
▶ PW :
a
▶ PW2 :
a
[##]Congratulation! you are now our member!
WELCOME SundiSodi
```

Application Execution

Login

- JoinUser HashMap
- LoginUser HaspMap
- ◆ Checking ID
- ◆ Checking Password

```
Are you our member? [YES/NO]
apple
[##] Wrong input.
Are you our member? [YES/NO]
yes
▶ ID :
apple
▶ PW :
a
Long time no see!)
```

Client1 : apple

Application Execution

HOME

Client1 : apple

```
===Login List===  
apple,  
apple 's warning : 0
```

```
HOME
```

```
-----  
Rule : Write like below.  
[#num]  
If you want to go HOME,  
Write [#HOME].
```

```
1      ROOM LIST  
2      1:1 CHAT  
3      MAKE NEW ROOM  
4      SEND LETTER  
5      BAD USER  
6      LOGOUT  
7      Exit  
8      Withdraw  
-----
```

► Input :

Application Execution

1:1 Chat

```
► Input :  
[##] [bag] has rejected 1:1 chat.  
#p2p bag  
[##] Let's chat - 1:1  
[##] You have requested 1:1 chat withbag  
► Input :  
[##] [bag] has rejected 1:1 chat.  
#p2p bag  
[##] Let's chat - 1:1  
[##] You have requested 1:1 chat withbag  
► Input :  
[##] [bag] has accepted 1:1 chat. Do you want to start?  
► Input:y  
  
[##] Let's start 1:1 chat.  
► Input :  
hihi  
[apple] hihi  
[bag] wow  
[bag] interesting!
```

Client1 : apple

```
^ [##] [apple] has requested 1:1 chat with you.  
Do you accept it?(y,n)  
► Input:y  
[##] send your message to apple  
[##] Let's start 1:1 chat.  
► Input :  
[apple] hihi  
wow  
[bag] wow  
interesting!  
[bag] interesting!
```

Client2 : bag

Application Execution

1:1 Chat

```
> Input :
#2
[##] Let's chat - 1:1
===Login List===
apple, cat, bag, dog,
apple 's warning : 1

Whom do you want to chat with? format : [#p2p name]
:
#p2p cat
[##] Let's chat - 1:1
[##] You have requested 1:1 chat withcat
> Input :
[##] That user does not exist or cannot talk 1:1
> Input :
#p2p bag
[##] Let's chat - 1:1
[##] You have requested 1:1 chat withbag
> Input :
[##] [bag] has rejected 1:1 chat.
```

Client1 : apple

```
Users: 3 people

Do you want to enter the room?
Then write like below.
[#IN <room name>]
or not, write [#HOME]]

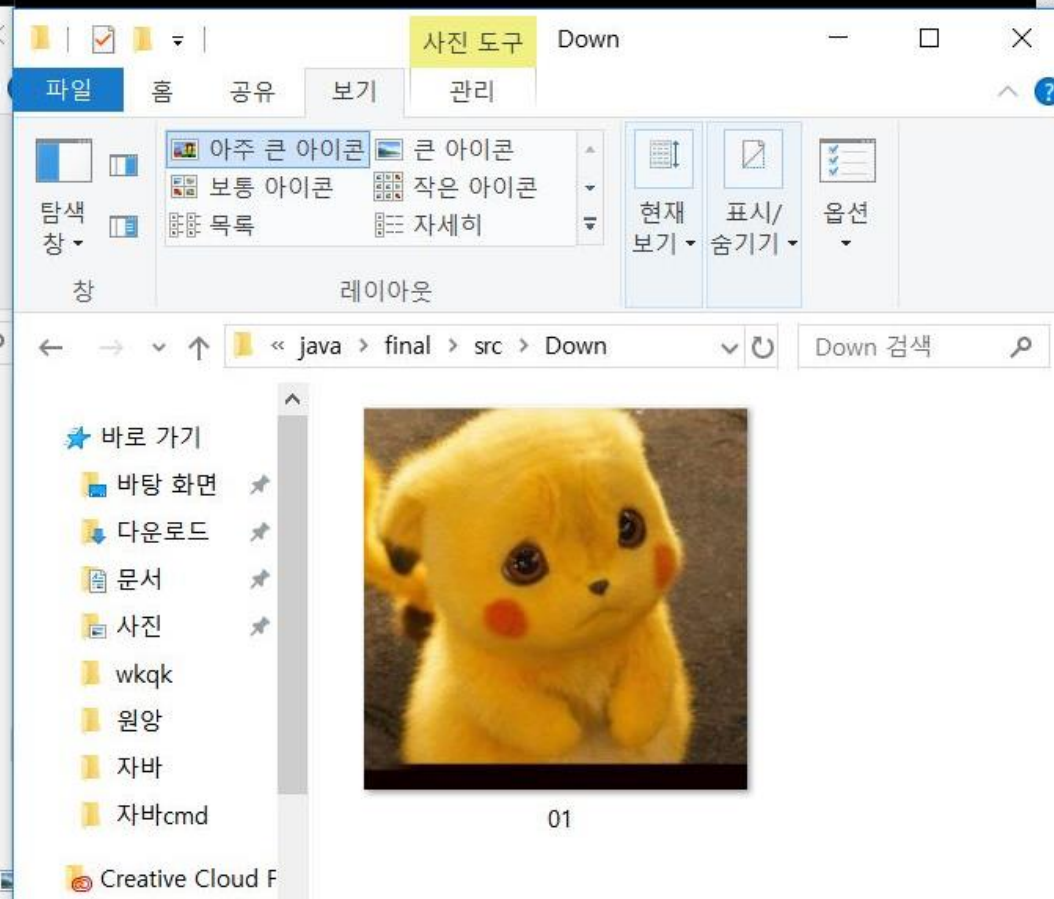
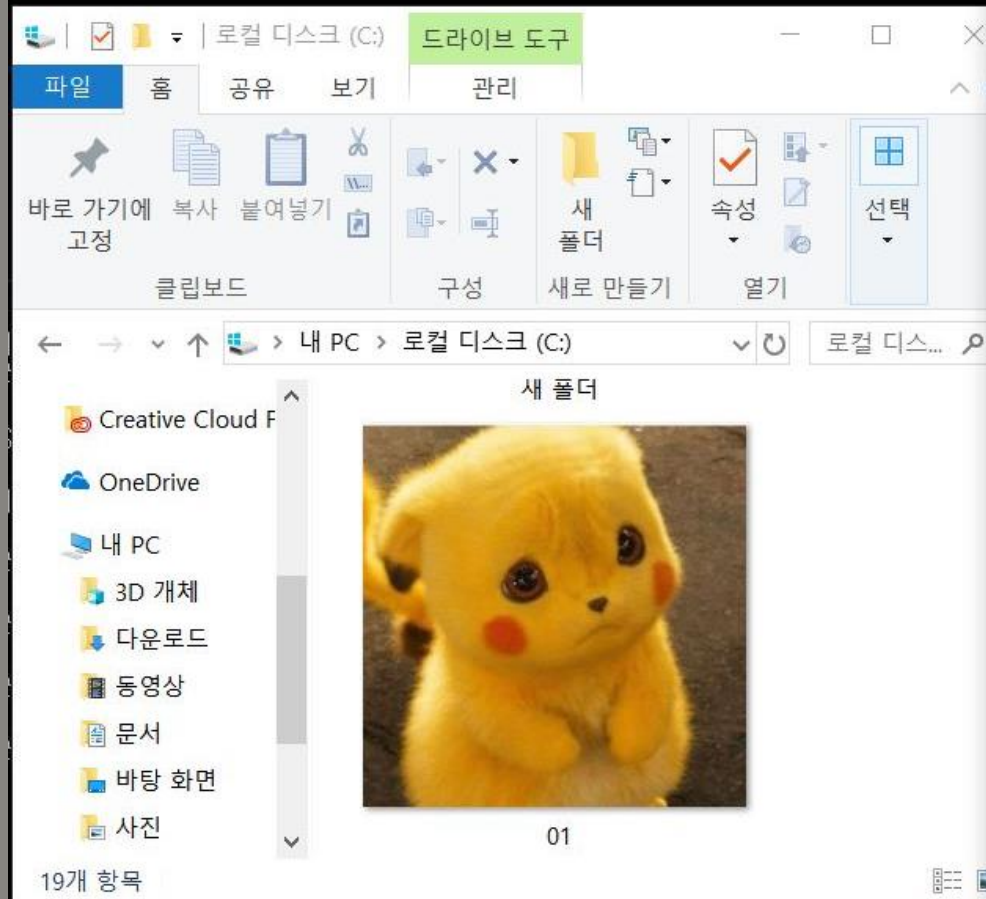
> Input :
[##] [apple] has requested 1:1 chat with you.
Do you accept it?(y,n)
> Input:n
[##] you have rejected 1:1 chat withapple
```

Client2 : bag

```
▶ Input :  
hihi  
[apple] hihi  
[bag] wow  
[bag] interesting!  
#FILE C:\01.jpg  
[##] Try to send file[C:\01.jpg] to [bag].
```

```
▶ Input :  
fileSender:192.168.171.1  
fileSender:C:\01.jpg  
=====> File Server sets up.  
=====> Start FileSend  
=====> FileSend Success
```

```
interesting!  
[bag] interesting!  
[##] [apple] tries to send a file[01.jpg].  
Do you accept it?(Y/N)  
▶ Input:y  
fileReceiver:192.168.171.1  
fileReceiver:127.0.0.1/01.jpg  
=====> Download File  
=====> Down01.jpg  
=====> Download File Success
```



Application Execution

Make new room

- GlobalMap HashMap

- ◆ Checking the name of Room.

```
▶ Input :  
#3 friend  
[##] 3. MAKE NEW ROOM  
  
[##] Room[friend] has been made  
▶ Input :  
===Room List===  
friend: (0 people)  
• Users: 5 people  
  
▶ Input :
```

Application Execution

Room List

Client1 : apple

- GlobalMap HashMap
- JoinUser HashMap

```
► Input :  
#1  
[##] 1. ROOM LIST  
  
===Room List===  
soccer: (0 people)  
fish: (0 people)  
friend: (0 people)  
animal: (0 people)  
⊙Users:5 people  
  
Do you want to enter the room?  
Then write like below.  
[#IN <room name>]  
or not, write [#HOME]]  
  
► Input :
```


Application Execution

- globalMap HashMap
- history HashMap

The image displays four terminal windows, each representing a different client in a multi-client chat application. The windows are titled '명령 프롬프트 - java multichat/MultiClient'.

Client3 : cat (top-left window):

```
animal: (0 people)
Users:5 people

Do you want to enter the room?
Then write like below.
[#IN <room name>]
or not, write [#HOME]]

> Input :
[#IN soccer
[##] You has entered Room (soccer)

HISTORY OF THIS CHATTING ROOM

Update the history of conversation
> Input :
[bag] heyguyshello
[bag] howareyou
[egg] who are you
[dog] happyhappyhappy
omg
[cat] omg
```

Client2 : bag (top-right window):

```
> Input :
[#IN soccer
[##] You has entered Room (soccer)

HISTORY OF THIS CHATTING ROOM

Update the history of conversation
> Input :
[##] [egg] has entered Room[soccer].
> Input :
[##] [dog] has entered Room[soccer].
> Input :
[##] [cat] has entered Room[soccer].
> Input :
heyguyshello
[bag] heyguyshello
howareyou
[bag] howareyou
[egg] who are you
[dog] happyhappyhappy
[cat] omg
```

Client4 : dog (bottom-left window):

```
Do you want to enter the room?
Then write like below.
[#IN <room name>]
or not, write [#HOME]]

> Input :
[#IN soccer
[##] You has entered Room (soccer)

HISTORY OF THIS CHATTING ROOM

Update the history of conversation
> Input :
[##] [cat] has entered Room[soccer].
> Input :
[bag] heyguyshello
[bag] howareyou
[egg] who are you
happyhappyhappy
[dog] happyhappyhappy
[cat] omg
```

Client5 : egg (bottom-right window):

```
Then write like below.
[#IN <room name>]
or not, write [#HOME]]

> Input :
[#IN soccer
[##] You has entered Room (soccer)

HISTORY OF THIS CHATTING ROOM

Update the history of conversation
> Input :
[##] [dog] has entered Room[soccer].
> Input :
[##] [cat] has entered Room[soccer].
> Input :
[bag] heyguyshello
[bag] howareyou
who are you
[egg] who are you
[dog] happyhappyhappy
[cat] omg
```

Inter

Do you want to enter the room?
Then write like below.
[#IN <room name>]
or not, write [#HOME]]

► Input : **#IN roomname**
#IN soccer
[##] You has entered Room (soccer)

HISTORY OF THIS CHATTING ROOM

Update the history of conversation

► Input :
[bag] heyguyshello
[bag] howareyou
[egg] who are you
[dog] happyhappyhappy
omg
[cat] omg

Client3 : cat

[##] You has entered Room (soccer)

HISTORY OF THIS CHATTING ROOM

Update the history of conversation

► Input :
[##] [egg] has entered Room[soccer].
► Input :
[##] [dog] has entered Room[soccer].
► Input :

[##] [cat] has entered Room[soccer].

► Input :
heyguyshello
[bag] heyguyshello
howareyou
[bag] howareyou
[egg] who are you
[dog] happyhappyhappy
[cat] omg

Client2 : bag

명령 프롬프트 - java multichat/MultiClient

Do you want to enter the room?
Then write like below.
[#IN <room name>]
or not, write [#HOME]]

► Input :
#IN soccer
[##] You has entered Room (soccer)

HISTORY OF THIS CHATTING ROOM

Update the history of conversation
► Input :
[##] [cat] has entered Room[soccer].

► Input :
[bag] heyguyshello
[bag] howareyou
[egg] who are you
happyhappyhappy
[dog] happyhappyhappy
[cat] omg

Client4 : dog

명령 프롬프트 - java multichat/MultiClient

Then write like below.
[#IN <room name>]
or not, write [#HOME]]

► Input :
#IN soccer
[##] You has entered Room (soccer)

HISTORY OF THIS CHATTING ROOM

Update the history of conversation

► Input :
[##] [dog] has entered Room[soccer].
► Input :
[##] [cat] has entered Room[soccer].

► Input :
[bag] heyguyshello
[bag] howareyou
who are you
[egg] who are you
[dog] happyhappyhappy
[cat] omg

Client5 : egg

SundiSodi

Interf

Do you want to enter the room?
Then write like below.
[#IN <room name>]
or not, write [#HOME]]

► Input :
#IN soccer
[##] You has entered Room (soccer)

HISTORY OF THIS CHATTING ROOM

Update the history of conversation

► Input :
[##] [cat] has entered Room[soccer].

[bag] hihi
[bag] hello
[bag] nice to meet u
[egg] hi
[egg] howareu
[cat] i love you
have a nice day~
[dog] have a nice day~

[##] Leave Room
[##] You has left Room[soccer].

===Room List===
soccer: (4 people)
fish: (0 people)
friend: (0 people)
animal: (0 people)
◎Users:5 people

Do you want to enter the room?
Then write like below.
[#IN <room name>]
or not, write [#HOME]]

Client4 : dog

Save Conversation

► Input :
#IN soccer
[##] You has entered Room (soccer)

HISTORY OF THIS CHATTING ROOM

[bag] hihi
[bag] hello
[bag] nice to meet u
[egg] hi
[egg] howareu
[cat] i love you
[dog] have a nice day~
[egg] bye
[bag] hungry
[apple] what do u want to eat
[bag] pizza
[apple] oh
[apple] i want some hamburger
[cat] metoo!

Update the history of conversation

► Input :
hihi
[dog] hihi

Client4 : dog

Interf

Whisper

```
명령 프롬프트 - java multichat/MultiClient
happy
[apple] happy
happyhappysmile
[apple] happyhappysmile
i lovejava
[apple] i lovejava
javajava
[apple] javajava
[bag] i lovecomputer
#WHISPER bag i love u
[whisper][apple] : i love u
```

Client1 : apple

```
명령 프롬프트 - java multichat/MultiClient
[whisper][apple] : i love JAVA
[##] [egg] has entered Room[soccer].
> Input :
[apple] happy
[apple] happyhappysmile
[apple] i lovejava
[apple] javajava
i lovecomputer
[bag] i lovecomputer
[whisper][apple] : i love u
#WHISPER egg apple loves me
[whisper][bag] : apple loves me
```

Client2 : bag

```
명령 프롬프트 - java multichat/MultiClient
> Input :
[apple] happy
[apple] happyhappysmile
[apple] i lovejava
[apple] javajava
[bag] i lovecomputer
```

Client4 : dog

```
명령 프롬프트 - java multichat/MultiClient
[dog] hihi
Update the history of conversation
> Input :
[apple] happy
[apple] happyhappysmile
[apple] i lovejava
[apple] javajava
[bag] i lovecomputer
[whisper][bag] : apple loves me
```

Client5 : egg

Application Execution

1:M Chat -Send Letter

- Enter the number of users you want to send letter.
- List the name of users.
- Input Message to send.

```
#4
[##] 4. SEND LETTER

[##] Whom do you want to send a letter?

===Login List===
apple, egg, cat, bag, dog,
apple 's warning : 0

[##] how many users do you want to send letters? [format : 3
]
3
[##] List the name of users [name name ...] :
egg cat bag
[##] What do you want to send? :
iloveyougoodnight
[##] Your letter is arrived well.

HOME
-----
Rule : Write like below.
[#num]
If you want to go HOME
```



```
명령 프롬프트 - java multichat/MultiClient
#4
[##] 4. SEND LETTER
[##] Whom do you want to send a letter?
==Login List==
apple, egg, cat, bag, dog,
apple's warning : 0
[##] how many users do you want to send letters? [format : 3
3
[##] List the name of users [name name ...] :
egg cat bag
[##] What do you want to send? :
iloveyoudoodnight
[##] Your letter is arrived well.
HOME
-----
Rule : Write like below.
[#num]
If you want to go HOME,
```

Client1 : apple

```
명령 프롬프트 - java multichat/MultiClient
[##] You has entered Room (soccer)
HISTORY OF THIS CHATTING ROOM
Update the history of conversation
> Input :
[bag] heyguyshello
[bag] howareyou
[egg] who are you
[dog] happyhappyhappy
omg
[cat] omg
[##] [apple] sent a letter to you! : iloveyoudoodnight
> Input :
```

Client3 : cat

[##]To whom?
==Login List==
apple, egg, cat, bag, dog

[##]How many?
Input : 3

[##]List name.
Input : egg cat bag

[##]What?
Input : iloveyoudoodnight

```
[##] [cat] has entered Room[soccer].
> Input :
[bag] heyguyshello
[bag] howareyou
[egg] who are you
happyhappyhappy
[dog] happyhappyhappy
[cat] omg
```

Client4 : dog

```
명령 프롬프트 - java multichat/MultiClient
> Input :
heyguyshello
[bag] heyguyshello
howareyou
[bag] howareyou
[egg] who are you
[dog] happyhappyhappy
[cat] omg
[##] [apple] sent a letter to you! : iloveyoudoodnight
> Input :
```

Client2 : bag

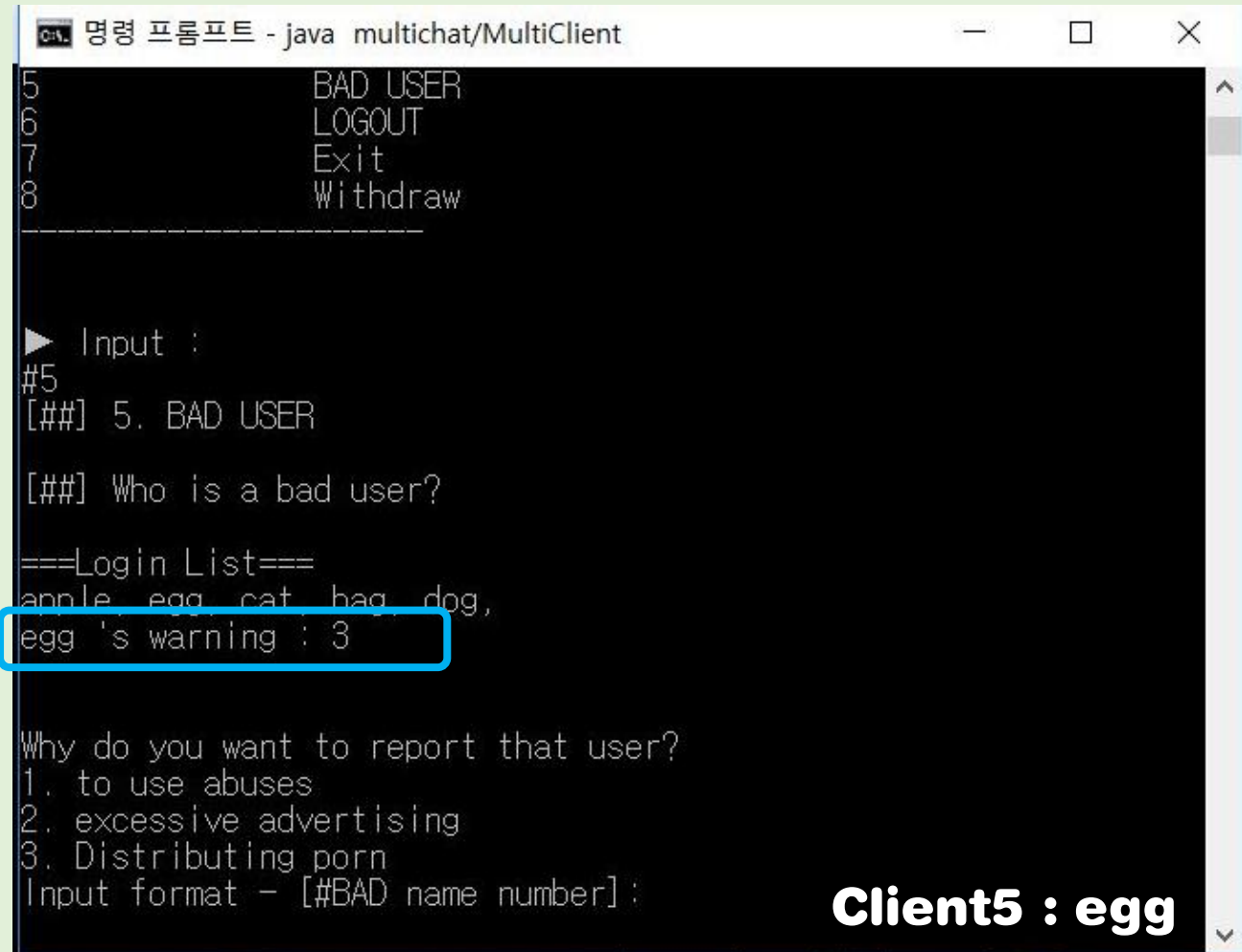
```
명령 프롬프트 - java multichat/MultiClient
[##] [cat] has entered Room[soccer].
> Input :
[bag] heyguyshello
[bag] howareyou
who are you
[egg] who are you
[dog] happyhappyhappy
[cat] omg
[##] [apple] sent a letter to you! : iloveyoudoodnight
> Input :
```

Client5 : egg

Application Execution

- LoginUser HashMap
- JoinUser HashMap
- cl HashMap

Bad User



```
명령 프롬프트 - java multichat/MultiClient
5          BAD USER
6          LOGOUT
7          Exit
8          Withdraw
-----
Input :
#5
[##] 5. BAD USER
[##] Who is a bad user?
===Login List===
apple, egg, cat, bag, dog,
egg 's warning : 3
Why do you want to report that user?
1. to use abuses
2. excessive advertising
3. Distributing porn
Input format - [#BAD name number]:
```

Client5 : egg

Interface / Execution

Bad User

명령 프롬프트 - java multichat/MultiClient

Update the history of conversation
▶ Input :
[##] [egg] is reported. [to use abuses]

[egg] omg
[egg] why i got warning sign
[egg] so sad
[bag] that's too bad
[##] [apple] is reported. [to use abuses]

[##] [egg] is reported. [to use abuses]
[##] [egg] is reported. [to use abuses]

[##] [bag] has left Room[soccer].
▶ Input :
[##] [egg] has left Room[soccer].
▶ Input :
[##] [egg] has entered Room[soccer].
▶ Input :
[##] [apple] has been withdrawn.

▶ Input :

명령 프롬프트 - java multichat/MultiClient

Then write like below.
[#IN <room name>
or not, write [#HOME]]

▶ Input :
#IN soccer
[##] You has entered Room (soccer)

HISTORY OF THIS CHATTING ROOM

[egg] omg
[egg] why i got warning sign
[egg] so sad
[bag] that's too bad
Update the history of conversation
▶ Input :
Good bye~

WELCOME SundiSodi
Are you our member? [YES/NO]
[##] [apple] has been withdrawn.

▶ Input :

Client1 : apple

Interface / Execution

```
HOME
-----
Rule : Write like below.
[#num]
If you want to go HOME,
Write [#HOME].

1          ROOM LIST
2          1:1 CHAT
3          MAKE NEW ROOM
4          SEND LETTER
5          BAD USER
6          LOGOUT
7          Exit
8          Withdraw
-----

▶ Input :
#6
[##] 6. LOGOUT

Good bye~

WELCOME SundiSodi
Are you our member? [YES/NO]
```

Logout

```
Client5 : egg

▶ Input :
#8
[##] 8. BYE SUNDISODI

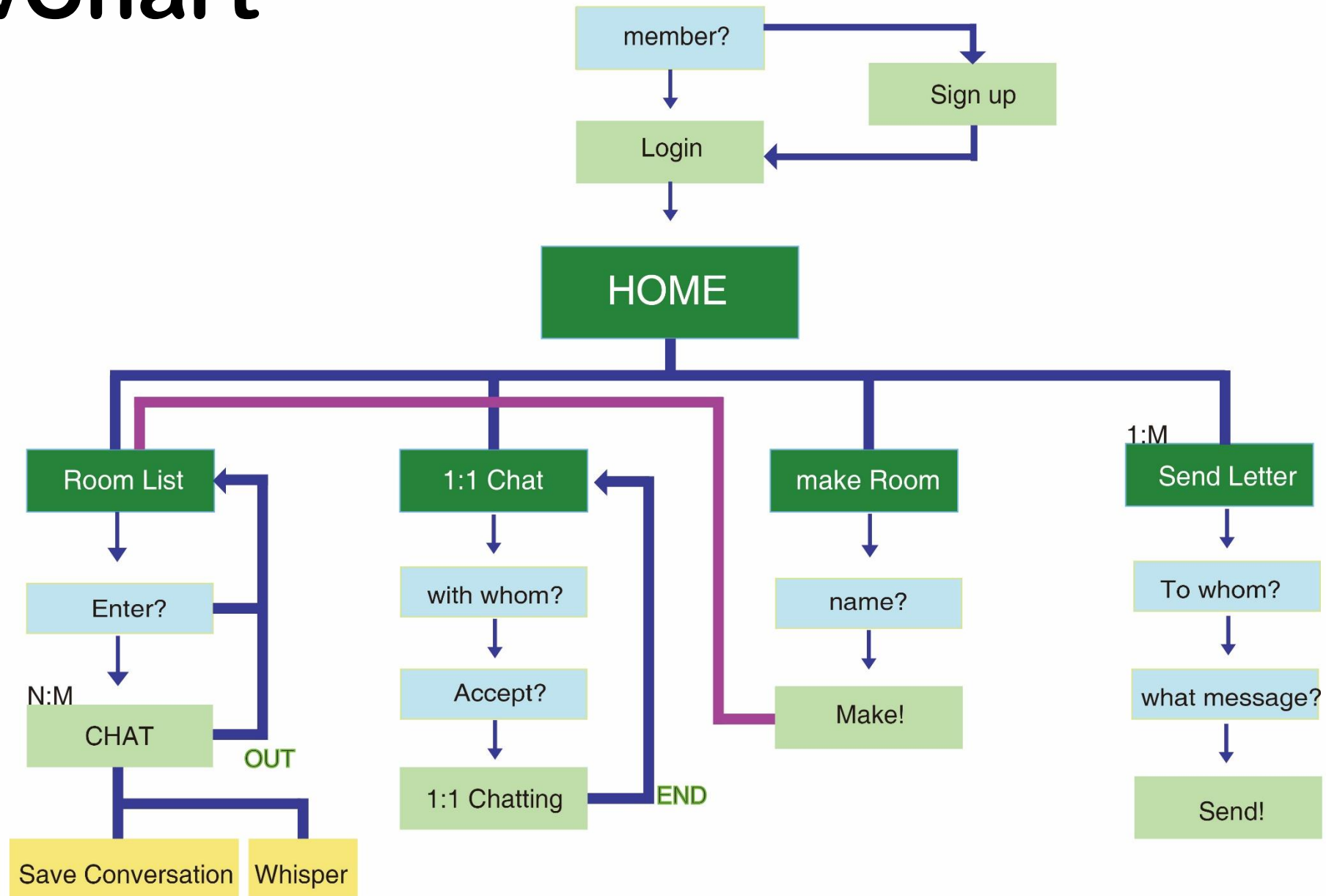
Good bye~

WELCOME SundiSodi
Are you our member? [YES/NO]
no
[##] Please be a member of <SundiSodi>
▶ ID :
egg
▶ PW :
e
▶ PW2 :
e
[##] Congratulation! you are now our member!
WELCOME SundiSodi

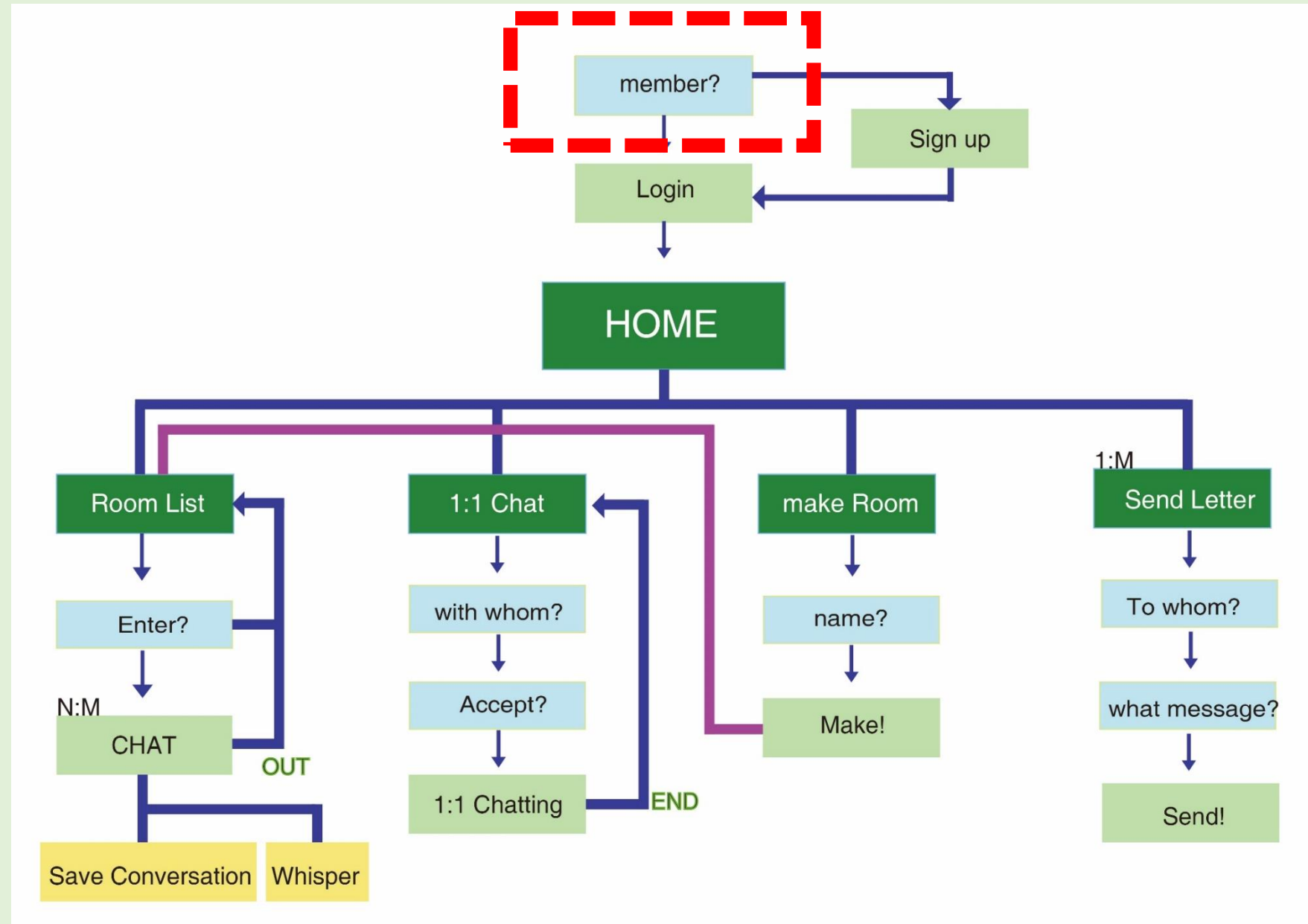
Are you our member? [YES/NO]
yes
▶ ID :
egg
▶ PW :
e
[##] ID is already in SundiSodi. Please put another ID.
Are you our member? [YES/NO]
```

Withdraw

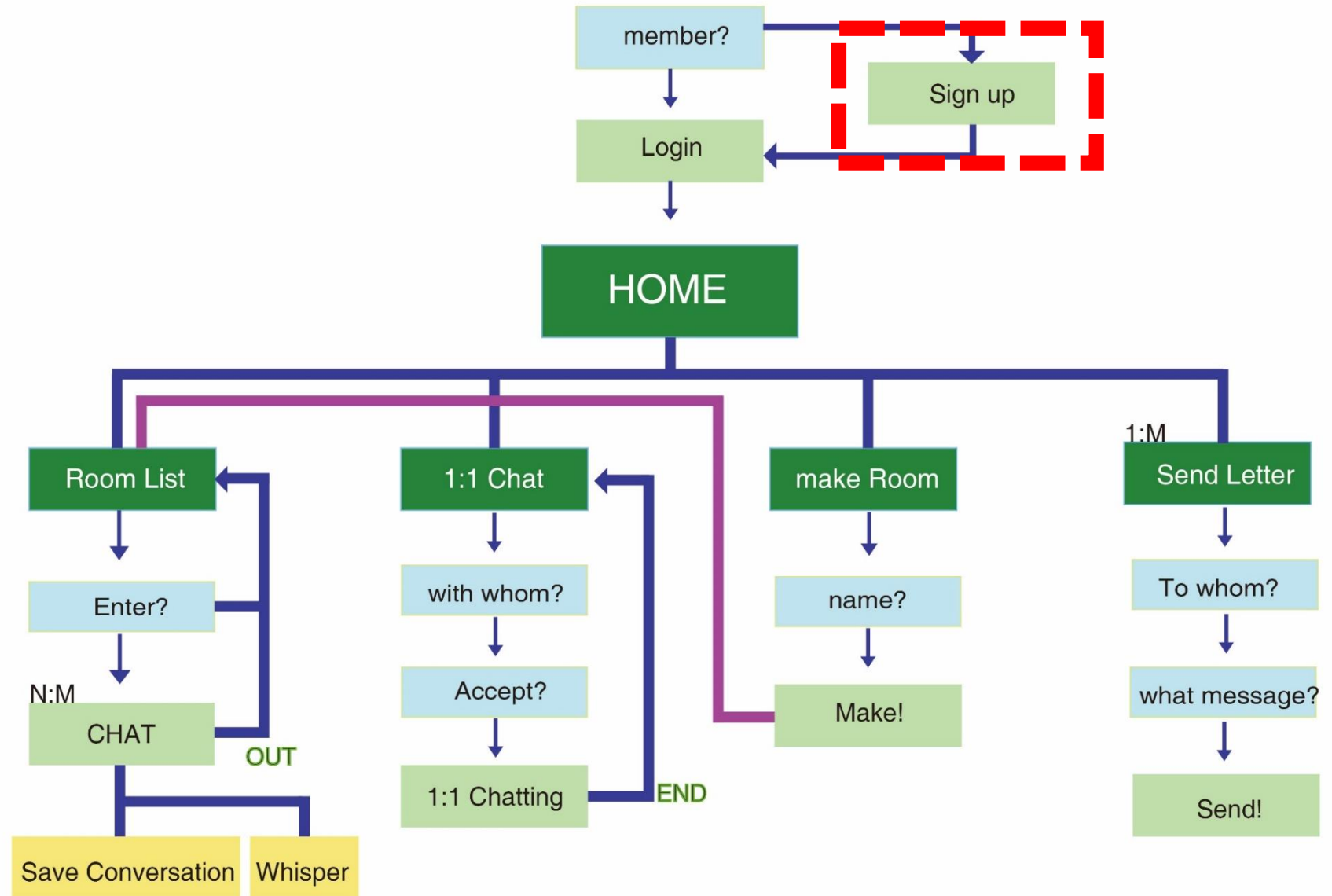
FlowChart



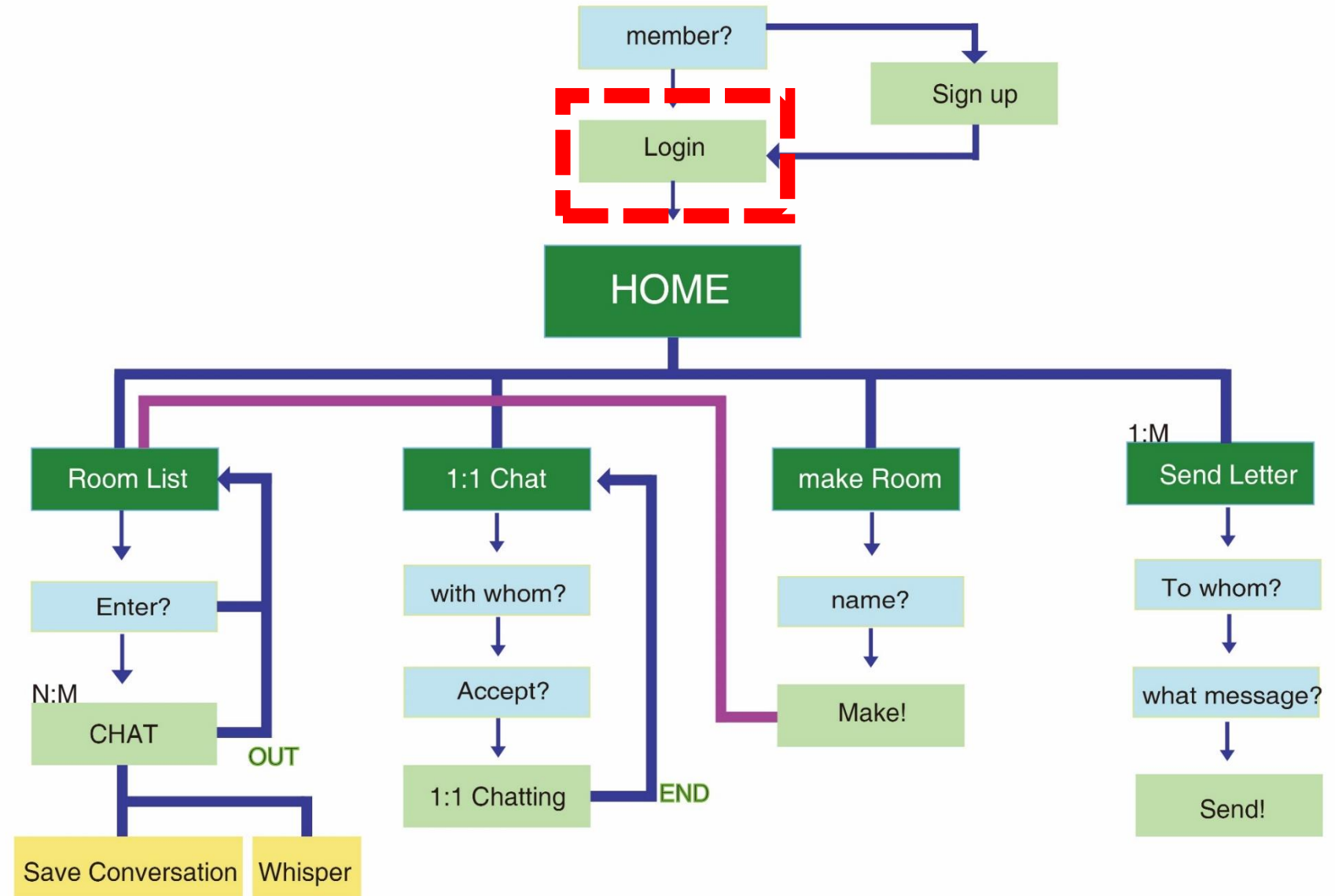
FlowChart



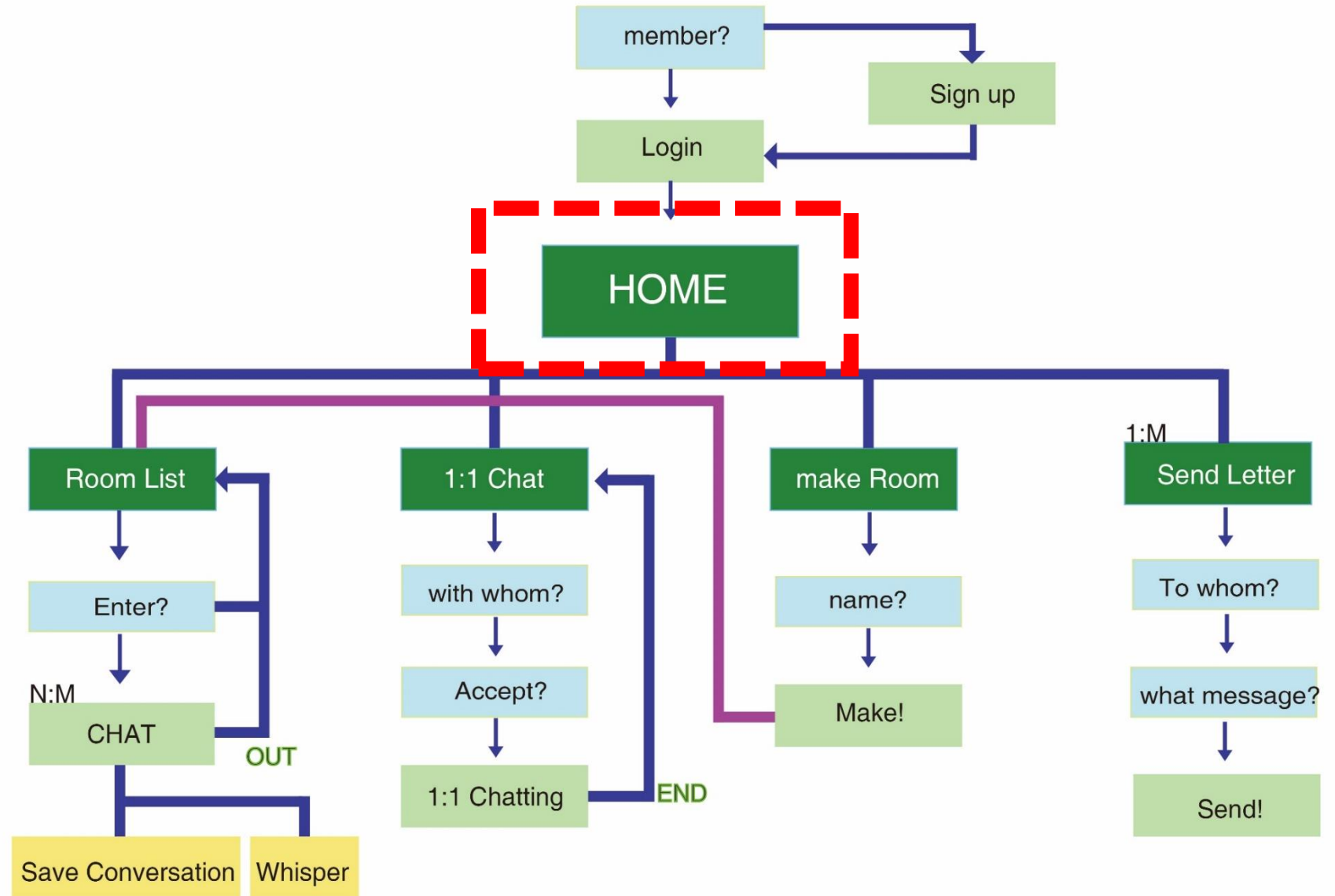
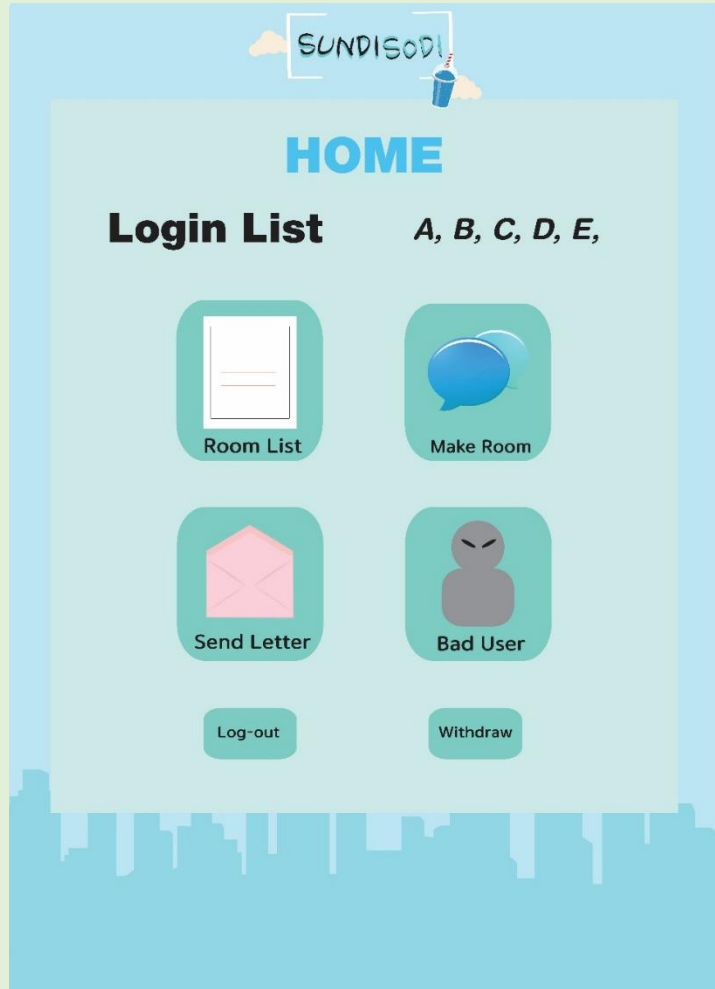
FlowChart



FlowChart



FlowChart

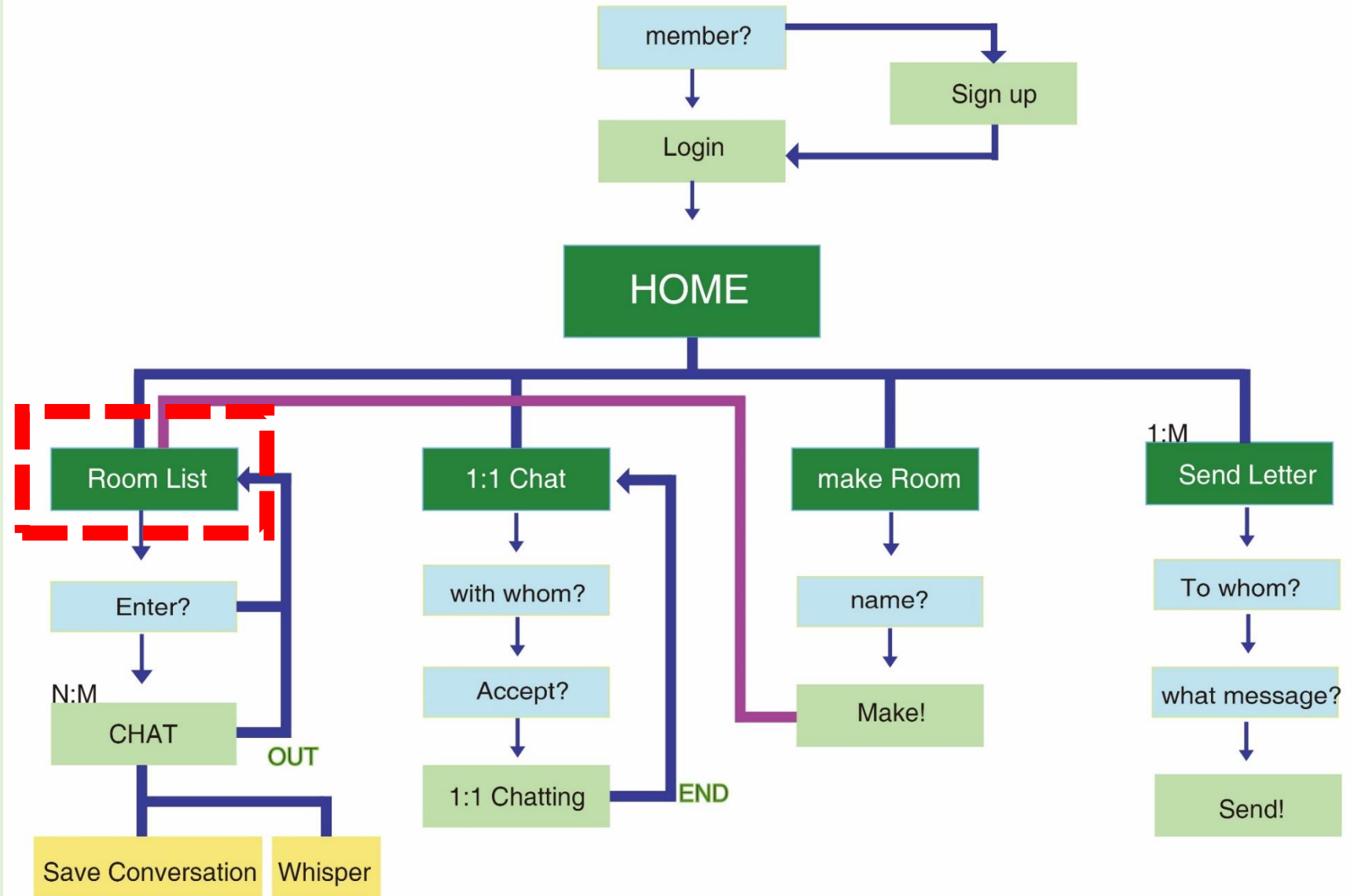


FlowChart

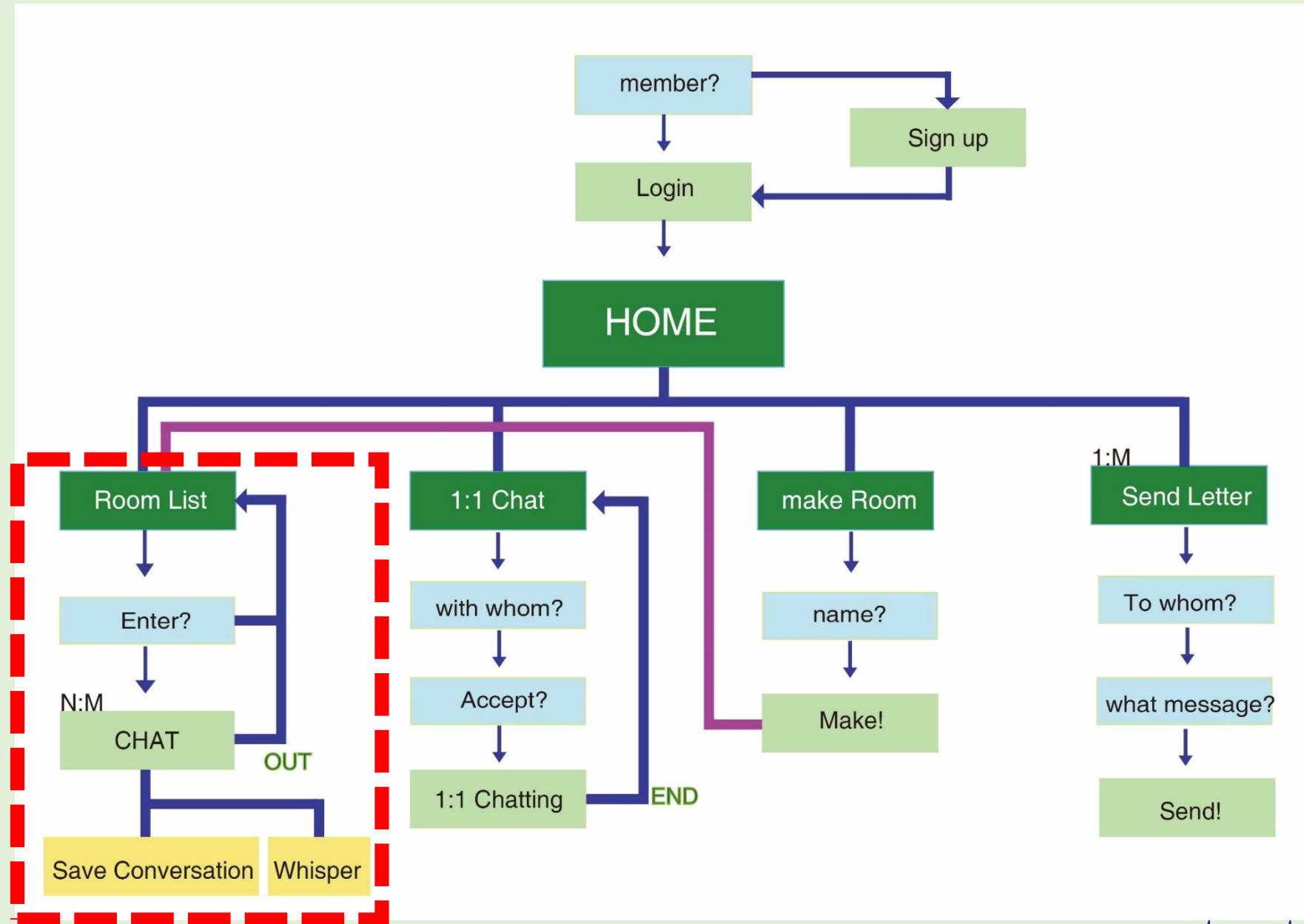
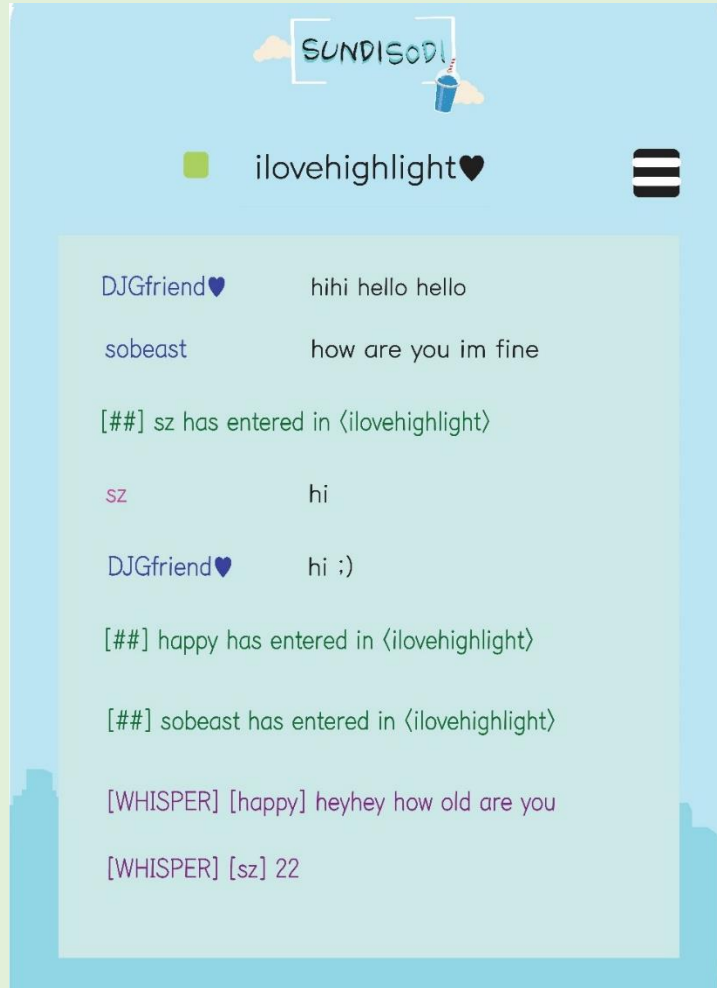


ROOM_List

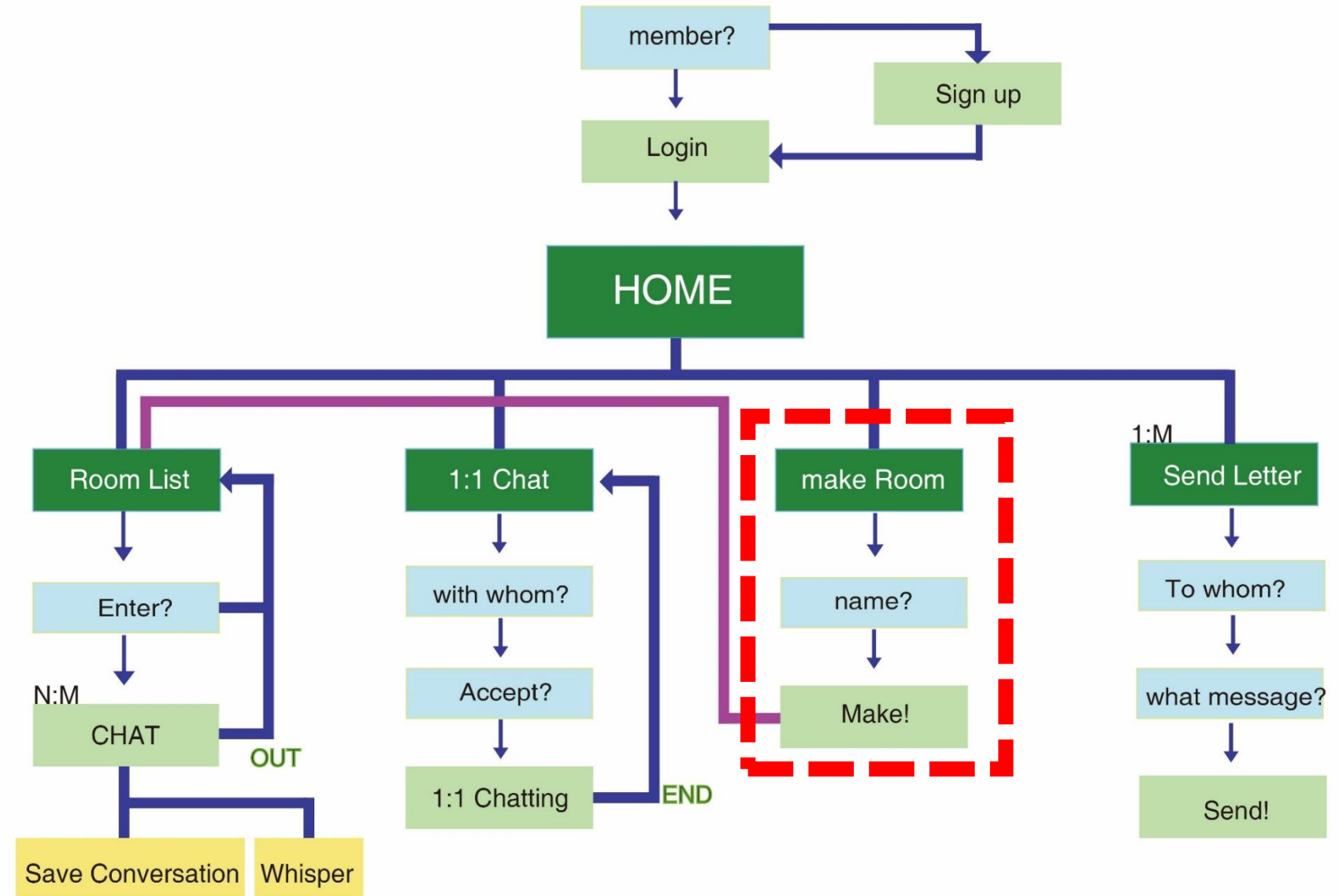
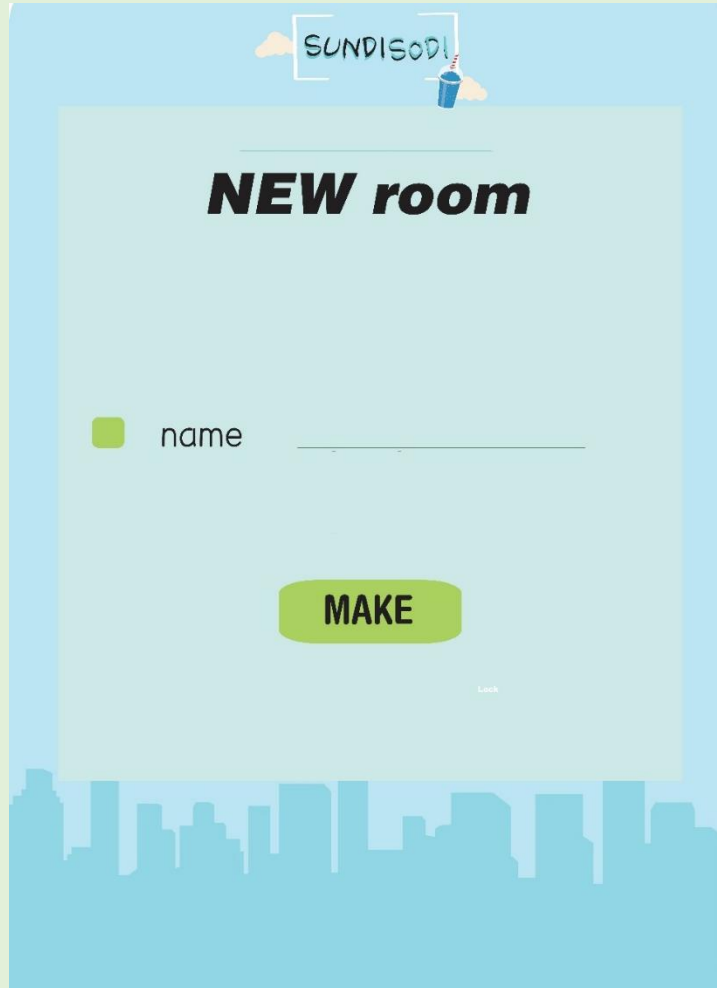
■ WeAreOne	8
■ ilovehighlight♥	3
■ KARMY	4
■ sunzisarang	1
■ HappyKitchen	1



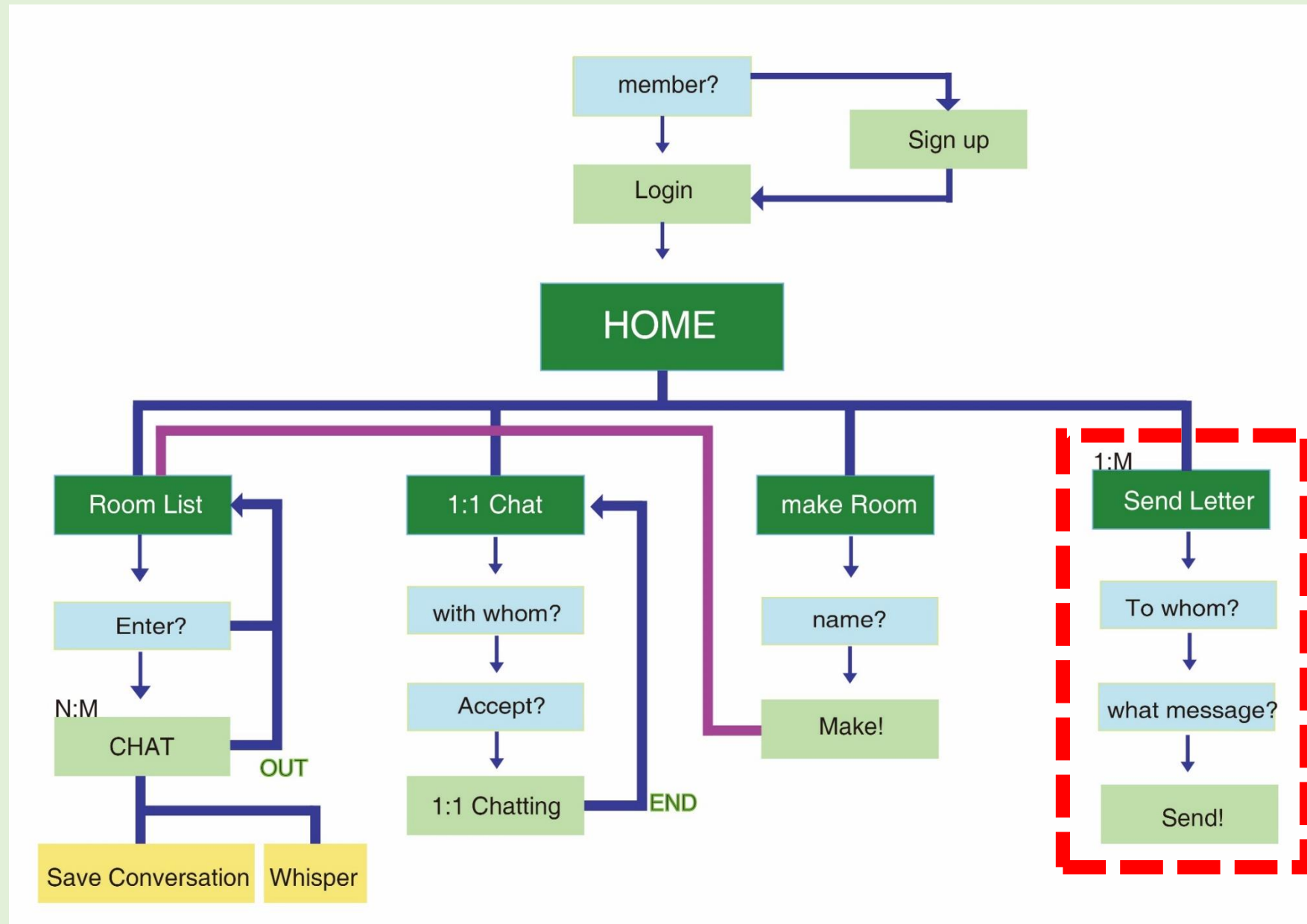
FlowChart



FlowChart



FlowChart



FlowChart



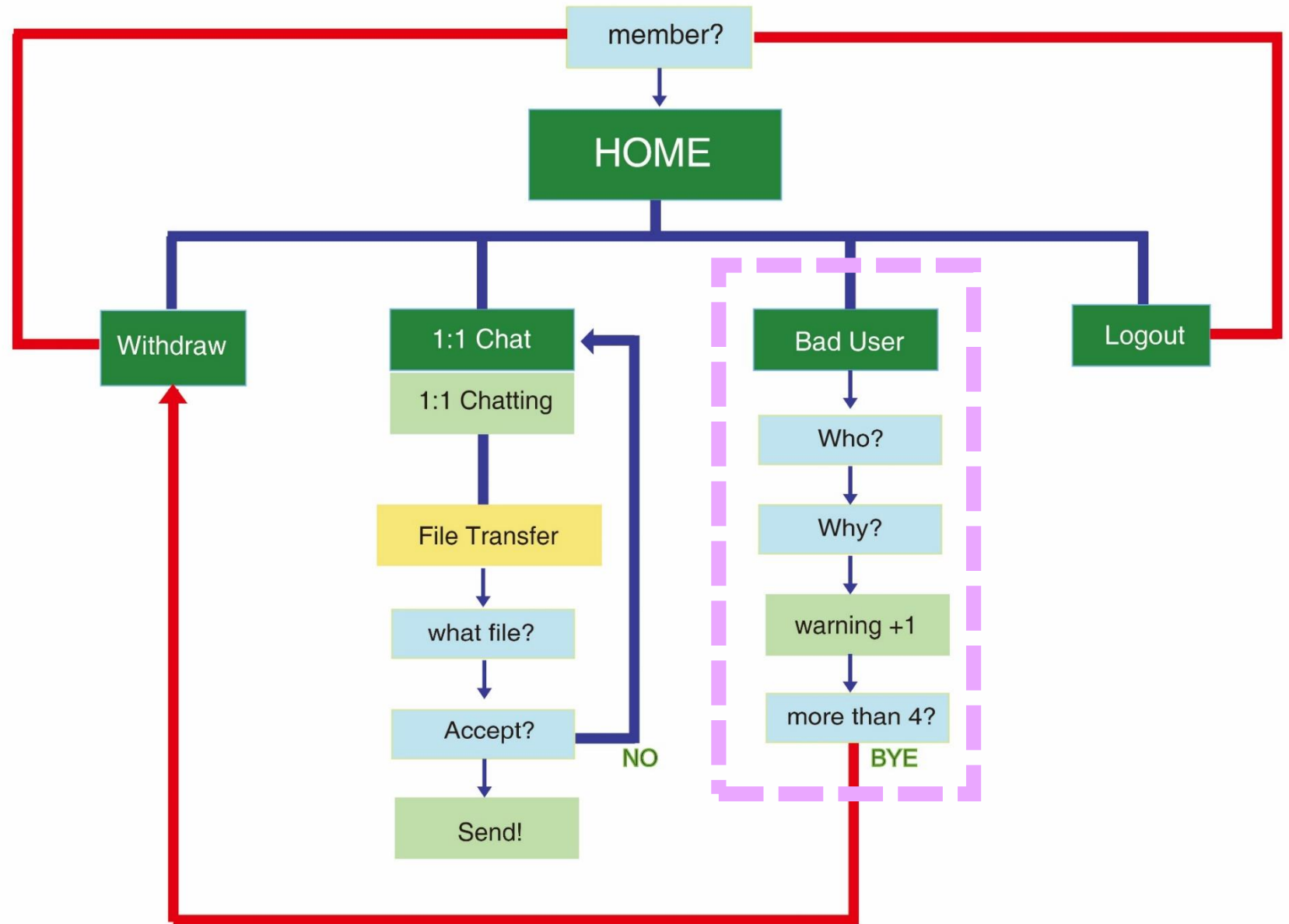
Bad User

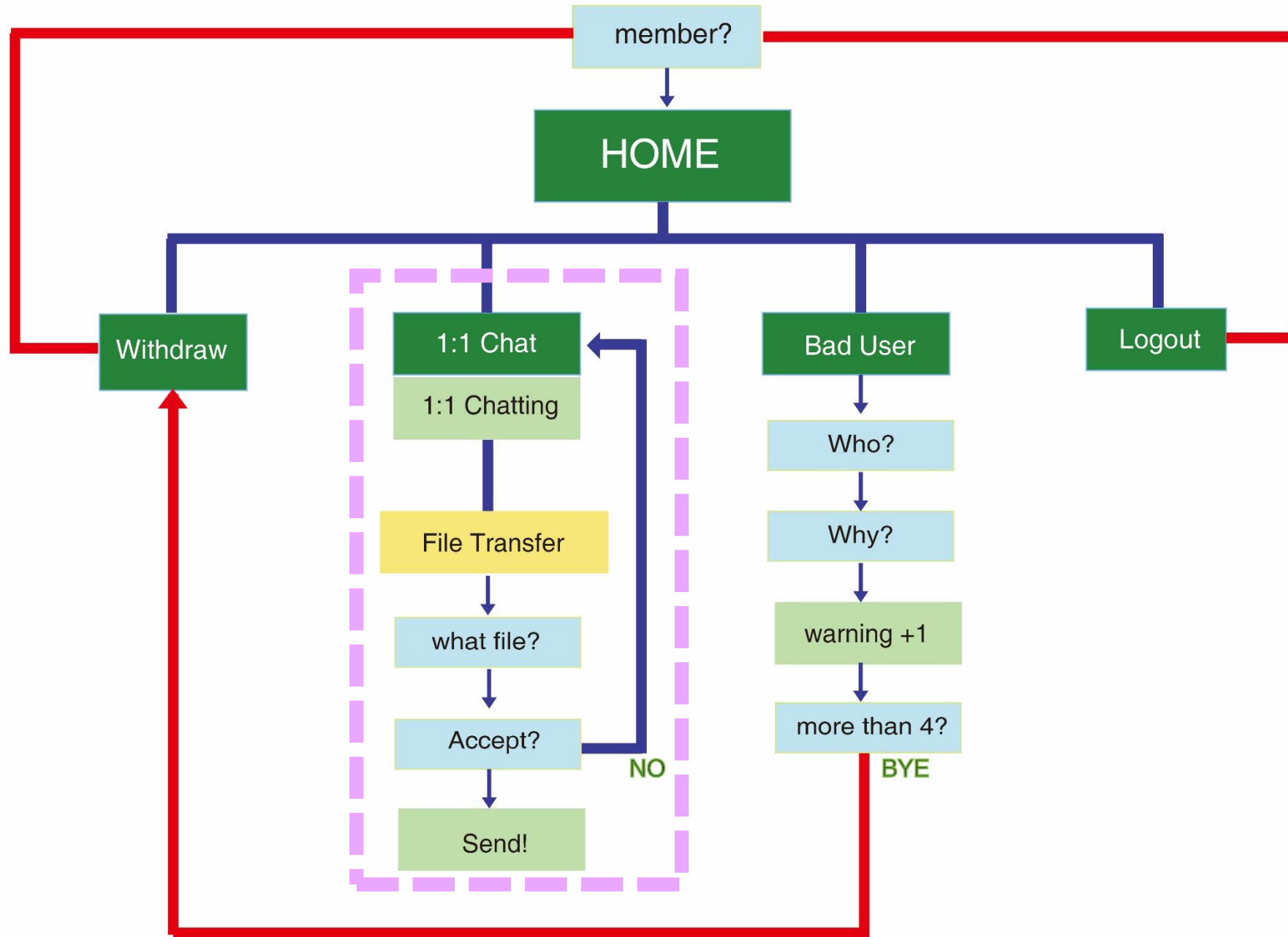
■ WHO?

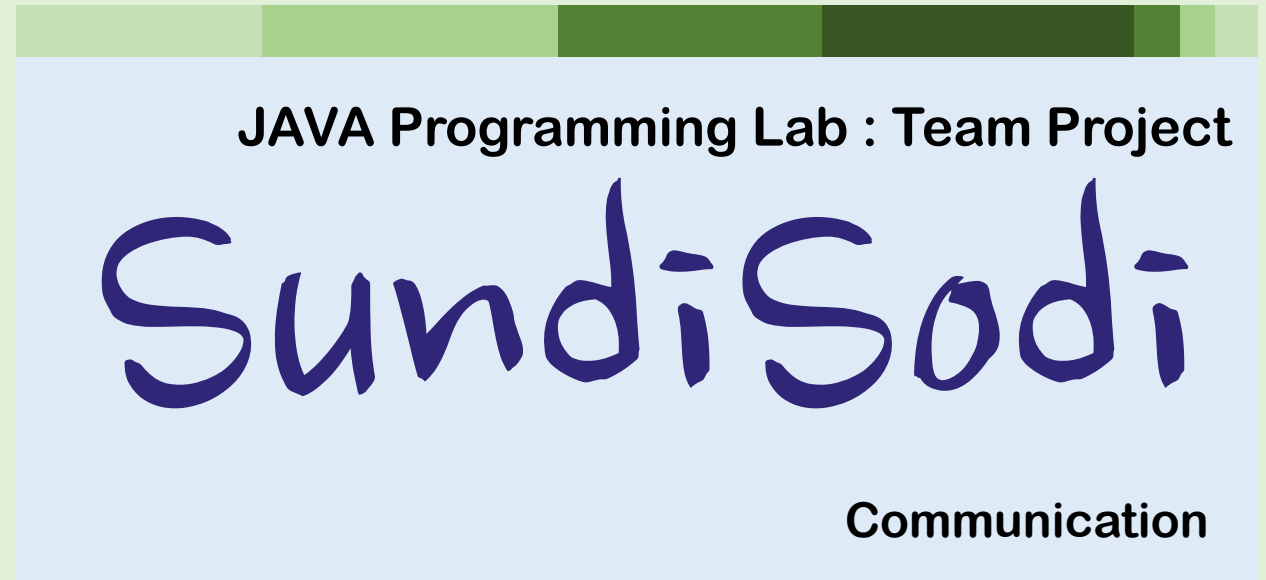
1. DJGfriend♥
2. sobeast
3. happyseonzi

■ WHY?

1. i hate
2. bad word
3. ad







TEAM 13 2016312860 KimSeonji
2017311456 KimSoyeong



THANK YOU

JAVA Programming Lab : Team Project

SundiSodi

Communication

TEAM 13 2016312860 KimSeonji
 2017311456 KimSoyeong