

# MỤC LỤC

DANH MỤC CÁC HÌNH .....	2
DANH MỤC CÁC BẢNG .....	3
1. Ngữ cảnh sử dụng phần mềm.....	4
2. Quá trình thực hiện.....	4
2.1 Thiết kế giao diện .....	4
2.2 Thiết kế lớp.....	6
2.2.1 Thiết kế lớp cho Server.....	6
2.2.1.1 Các lớp được sử dụng trong chương trình cho Server .....	6
2.2.1.2 Mô tả các phương thức của một lớp cho Server.....	7
2.2.2 Thiết kế lớp cho Client .....	15
2.2.2.1 Các lớp được sử dụng trong chương trình cho Client .....	15
2.2.2.2 Mô tả các phương thức của một lớp cho Client .....	16
3. Phân công công việc.....	29
KẾT LUẬN .....	30
TÀI LIỆU THAM KHẢO .....	31

# DANH MỤC CÁC HÌNH

Hình 1.1: Màn hình Server ..... 4

Hình 1.2: Màn hình Client..... 5

Hình 1.3: Màn hình History..... 6

## DANH MỤC CÁC BẢNG

Bảng 2.1 Thiết kế giao diện.....	6
Bảng 2.2 Danh mục các lớp được sử dụng trong chương trình cho Server .....	7
Bảng 2.3 Bảng mô tả các phương thức trong lớp Client cho Server.....	8
Bảng 2.4 Bảng mô tả các phương thức trong lớp Database cho Server.....	10
Bảng 2.5 Bảng mô tả các phương thức trong lớp ServerThread cho Server.....	11
Bảng 2.6 Bảng mô tả các phương thức trong lớp ServerSocket cho Server .....	13
Bảng 2.7 Bảng mô tả các phương thức trong lớp ServerFrame cho Server.....	15
Bảng 2.8 Danh mục các lớp được sử dụng trong chương trình cho Client.....	16
Bảng 2.9 Bảng mô tả các phương thức trong lớp Message cho Client .....	17
Bảng 2.10 Bảng mô tả các phương thức trong lớp Download cho Client.....	18
Bảng 2.11 Bảng mô tả các phương thức trong lớp Upload cho Client .....	20
Bảng 2.12 Bảng mô tả các phương thức trong lớp History cho Client .....	23
Bảng 2.13 Bảng mô tả các phương thức trong lớp SocketClient cho Client .....	29
Bảng 3.1 Bảng mô tả phân công công việc .....	299

# NỘI DUNG

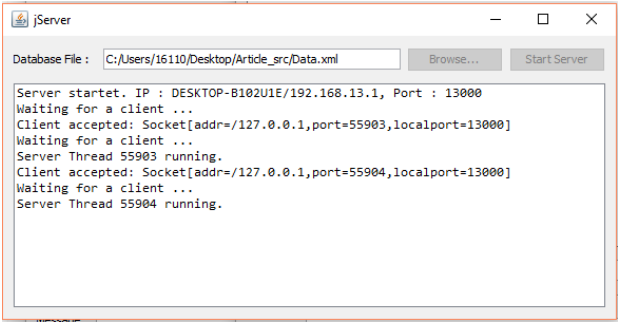
## 1. Ngữ cảnh sử dụng phần mềm

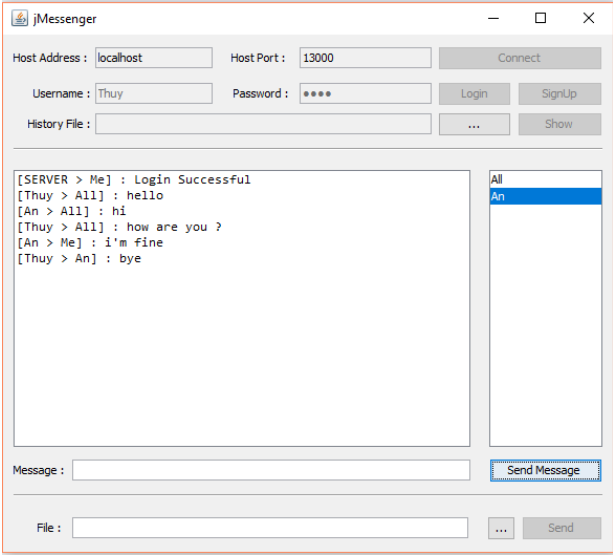
Phần mềm được phát triển chạy trên nền hệ điều hành Windows giúp người dùng có thể kết nối tới một server máy chủ do người dùng nhập vào, từ đó tạo ra một môi trường giúp họ có thể trò chuyện, trao đổi tài liệu qua lại với nhau, bên cạnh đó, chúng ta còn có thể gửi tin nhắn tới tất cả người đang online.

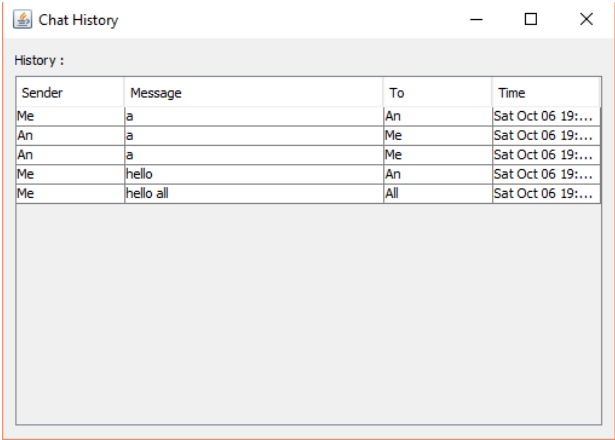
## 2. Quá trình thực hiện

### 2.1 Thiết kế giao diện

- Phần mềm được thiết kế để chạy trên hệ điều hành Windows
- Ngôn ngữ được sử dụng: Java
- Thư viện được sử dụng để thiết kế giao diện: Swing
- Môi trường lập trình: Eclipse
- Hướng lưu trữ dữ liệu: Lưu vào file định dạng \*.xml

TT	Màn hình/Cửa sổ/Dialog	Người thiết kế & giải thích ngắn gọn	Mục đích chính của màn hình
1	<div><p>Màn hình cho Server</p></div> <p>Hình 1.1: Màn hình Server</p>	<p>Ngô Công An</p> <ul style="list-style-type: none"><li>- Button chọn đường dẫn tới file chứa dữ liệu người dùng (*.xml) bao gồm tài khoản và mật khẩu người dùng</li></ul>	<p>Tạo một giao diện đơn giản cho người quản lý server và thể hiện được các trạng thái của server khi đang chạy.</p>

		<ul style="list-style-type: none"> <li>- Button chạy Server</li> <li>- Panel thể hiện các trạng thái của chương trình (trạng thái kết nối, client đăng nhập, ...).</li> </ul>	
2	<p>Màn hình chat cho Client</p>  <p>Hình 1.2: Màn hình Client</p>	<p>Đào Xuân Thủy</p> <ul style="list-style-type: none"> <li>- Cho phép người dùng nhập địa chỉ máy chủ, cổng kết nối và có thể kết nối tới Server.</li> <li>- Đăng nhập bằng tài khoản cá nhân hoặc đăng ký tài khoản.</li> <li>- Chọn file dùng để lưu lại lịch sử chat của mình.</li> <li>- Button xem lại lịch sử chat.</li> </ul>	<p>Tạo một giao diện chat đơn giản, dễ sử dụng, gồm đầy đủ các tính năng cần thiết và dễ dàng mở rộng chương trình.</p>

		<ul style="list-style-type: none"> <li>- Panel giao diện chat.</li> <li>- Cửa sổ chọn người nhận (chat với tất cả hoặc một cá nhân).</li> <li>- Khung nhập tin nhắn để gửi đi.</li> <li>- Chọn file để gửi file cho người nhận.</li> </ul>	
3	Màn hình hiển thị lịch sử chat   <p>Hình 1.3: Màn hình History</p>	Đào Xuân Thủy  - Tạo một khung hiển thị nội dung của lịch sử tin nhắn gồm: Người gửi, người nhận, nội dung tin nhắn và thời gian gửi tin.	Xem lịch sử chat của khách hàng khi ấn vào button “Show” tại cửa sổ chat.

Bảng 2.1 Thiết kế giao diện

## 2.2 Thiết kế lớp

### 2.2.1 Thiết kế lớp cho Server

#### 2.2.1.1 Các lớp được sử dụng trong chương trình cho Server

TT	Tên Lớp	Người thiết kế	Mục đích thiết kế
1	Client	Ngô Công An	Tạo một đối tượng client chứa những thông tin cần thiết của một tin nhắn được gửi đi hoặc nhận về bao gồm: người gửi, người nhận, nội dung và thời gian gửi để lưu trữ thông tin tin nhắn. Truyền yêu cầu kiểu Message từ Giao diện vào đối tượng SocketClient để thực thi chức năng cho chương trình (đăng nhập, đăng ký, upload, ...).
2	Database	Ngô Công An	Tạo một đối tượng Database lấy thuộc tính filePath. Mục đích của đối tượng là kiểm tra đăng nhập và tạo tài khoản
3	ServerThread	Ngô Công An	Tạo đối tượng ServerThread lấy những thuộc tính như: SocketServer, socket, ObjectInputStream, ObjectOutputStream, ServerFrame. Mục đích là tạo Thread để xử lý nhận, gửi tin nhắn
4	SocketServer	Ngô Công An	Tạo đối tượng SocketServer lấy những thuộc tính như: ServerThread, Thread, ServerFrame, Database. Mục đích là nơi tạo kết nối cho các client, tạo và đóng thread, nhận và gửi các tin nhắn của các client
5	ServerFrame	Ngô Công An	Tạo đối tượng ServerFrame lấy những thuộc tính như: SocketServer, Thread, filePath, JFileChooser. Mục đích là xây dựng giao diện server (jTextFiled, jButton, jTextArea, jScrollPane, jLabel)

Bảng 2.2 Danh mục các lớp được sử dụng trong chương trình cho Server

#### 2.2.1.2 Mô tả các phương thức của một lớp cho Server

<b>T T</b>	<b>Tên phư ơng thứ c</b>	<b>Tên các sinh viên phụ trách viết</b>	<b>Mục đích chính của phương thức</b>	<b>Input</b>	<b>Ou tpu t</b>	<b>Mã giả</b>	<b>Tên file có sử dụng phương thức</b>
	Clie nt	Ngô Công An	Tạo các đối tượng của một tin nhắn ( type,sender,content,recipient)	Type,Sender,Content,Recipient		This.type = type, this.sender = sender,this .content = content , this.recipei nt = recipient	SocketServ er (46,260)
	toSt ring	Ngô Công An	Trả về một chuỗi kiểu String chứa các thông tin của một tin nhắn như Type, Sender, Recipient, Content		Stri ng	Return về chuỗi	

Bảng 2.3 Bảng mô tả các phương thức trong lớp Client cho Server

<b>T T</b>	<b>Tên phươ ng thức</b>	<b>Tên các sinh viên</b>	<b>Mục đích chính của phương thức</b>	<b>Input</b>	<b>Outp ut</b>	<b>Mã giả</b>	<b>Tên file có sử dụng phương thức</b>
----------------	-------------------------------------	--------------------------------------	---	--------------	--------------------	---------------	--



		<b>phụ trách viết</b>					
1	Datab ase	Ngô Công An	Tạo ra đối tượng filePath	filePat h		This.filePath = filePath	SocketSer ver (86,106)
2	userE xit	Ngô Công An	Kiểm tra user đang được tạo có bị trùng với các user đã được tạo trong filePath	userna me	True or false	If (getTagValue ("username", eElement).eq ual(username ) {return true } return false	Database (51)  SocketSer ver (214
3	Check Login	Ngô Công An	Kiểm tra user và password lúc đăng nhập có đúng với giá trị trong data	userna me, passwo rd	True or false	If(!userExit(u sername)) return false  Else {return true}	
4	addUs er	Ngô Công An	Tạo newuser và newpassword sau đó lưu vào filePath	Userna me, passwo rd		Input username and password output filePath	
5	getTa gValu e	Ngô Công An	Lấy các tài khoản có trong filePath để đăng nhập	sTag, eElem ent	Giá trị trong	Input một element,	Database (36, 66)

					filePath	output giá trị của element	
--	--	--	--	--	----------	----------------------------	--

Bảng 2.4 Bảng mô tả các phương thức trong lớp Database cho Server

TT	Tên phương thức	Tên các sinh viên phụ trách viết	Mục đích chính của phương thức	Input	Output	Mã giả	Tên file có sử dụng phương thức
1	send	Ngô Công An	Gửi đi tin nhắn giữa các client	Msg		streamOut.writeObject streamOut.flush	SocketServer (180,186, 191, 202, 203, 208, 218, 219, 225, 230, 237, 241, 249, 253, 263, 270)
2	getID	Ngô Công An	Trả về giá trị ID vừa gửi message		ID	Output ID	

3	run	Ngô Công An	Chạy thread cho hiển thị client đang hoạt động			While(true) client msg = (client)streamIn.readObject	
4	open	Ngô Công An	Tạo môi trường để các client hoạt động			Mở kết nối Thread	SocketServer (319)
5	close	Ngô Công An	Đóng thread làm việc khi client ngưng hoạt động			Đóng kết nối Thread	SocketServer (66,68,70,301)

Bảng 2.5 Bảng mô tả các phương thức trong lớp ServerThread cho Server

<b>T T</b>	<b>Tên phương thức</b>	<b>Tên các sinh viên phụ trách viết</b>	<b>Mục đích chính của phương thức</b>	<b>Input</b>	<b>Output</b>	<b>Mã giả</b>	<b>Tên file có sử dụng phương thức</b>
1	SocketServer	Ngô Công An	Tạo server socket, tạo số thread có thể hoạt động	Frame		If(port=server.getLocalPort) { start}  Else { RetryStart }	ServerFrame (101, 112)  SocketServer (90, 110)

2	run	Ngô Công An	Chạy server và chờ các kết nối từ client	Frame, port		While(thread !=null) addThread	
3	start	Ngô Công An	Tạo thread cho các client hoạt động			If(thread == null) { start}	SocketServer (93,113,320)
4	stop	Ngô Công An	Ngắt thread			If(thread !=null){ stop }	SocketServer (53,307) ServerFrame (112)
5	findClient	Ngô Công An	Tìm các client và kiểm tra xem client có trong dữ liệu	ID	Trả về giá trị ID	Tìm client và output ID	
6	handle	Ngô Công An	Kiểm tra khi client kết nối nếu đúng tạo thread cho client hoạt động, nhận và gửi đi các tin nhắn của client trong thread (client tới client hoặc client tới tất cả)	ID, msg		Kiểm tra kết nối và các hoạt động gửi tin nhắn	

7	Announce	Ngô Công An	Gửi tin nhắn đến tất cả client trong thread	Type, sender, content		Input client output ID	SocketServer (167, 176, 189, 206)
8	findUserThread	Ngô Công An	Tìm client trong thread	usr	Vị trí của client trong thread	If(client[i].username.equals(usr)) thì trả về giá trị của client	SocketServer (172, 192, 200, 222, 227, 228, 231, 270)
9	remove	Ngô Công An	Xóa client khỏi thread và đóng thread	ID		Input ID kiểm tra client và xóa client	SocketServer (52,168)
10	addThread	Ngô Công An	Khi có client kết nối sẽ cho kiểm tra và tạo ra thread để client hoạt động	socket		If(clientCount < clients.length) thì cho client kết nối  Ngược lại thì k cho kết nối	SocketServer (127)

Bảng 2.6 Bảng mô tả các phương thức trong lớp ServerSocket cho Server

<b>T T</b>	<b>Tên phương thức</b>	<b>Tên các sinh viên phụ trách viết</b>	<b>Mục đích chính của phương thức</b>	<b>Input</b>	<b>Output</b>	<b>Mã giả</b>	<b>Tên file có sử dụng phương thức</b>
1	ServerFrame	Ngô Côn g An	Tạo giao diện server			fileChooser = new jFileChooser	
	initComponents	Ngô Côn g An	Tạo button bắt sự kiện như Browse, Start server			Input (JButton, Jtext, JLabel, JScrollPane) output là màn hình server	
3	jButton1ActionPerformed	Ngô Côn g An	Khi kết nối server thành công thì cho ấn 2 button Browse và Start server	Evt		Server = new SocketServer(this) và cho tắt các nút nhấn	

4	RetryStart	Ngô Côn g An	Nếu chưa có server thì start server mới	port		If(server!=null){ stop}  Else{ server = new SocketServer(98, 132) this,port)}	Soc ketS erve r
5	jButton2ActionPerformed	Ngô Côn g An	Nếu chưa chọn filePath thì ấn nút Start server	Evt		If(file!=null){ filePath = file.getPath}	

Bảng 2.7 Bảng mô tả các phương thức trong lớp ServerFrame cho Server

## 2.2.2 Thiết kế lớp cho Client

### 2.2.2.1 Các lớp được sử dụng trong chương trình cho Client

TT	Tên Lớp	Người thiết kế	Mục đích thiết kế
1	Message	Đào Xuân Thủy	Chứa những thông tin cần thiết của một tin nhắn được gửi đi hoặc nhận về, bao gồm: người gửi, người nhận, nội dung và thời gian gửi để lưu trữ thông tin tin nhắn. Truyền yêu cầu kiểu Message từ Giao diện vào đối tượng SocketClient để thực thi chức năng cho chương trình (đăng nhập, đăng ký, upload, ... ).
2	Download	Đào Xuân Thủy	Lấy những thuộc tính từ giao diện chat như ServerSocket, port, Socket và các thuộc tính khác: địa chỉ lưu file, FileOutputStream, InputStream lấy từ socket. Mục đích của đối tượng dùng để thực hiện quá trình tải một tệp từ giao diện chat về máy.

3	Upload	Đào Xuân Thủy	Lấy những thuộc tính như địa chỉ máy chủ, port, File và đối tượng Giao diện chat dùng để thay đổi giao diện sau khi chạy code. Mục đích của đối tượng dùng để thực hiện quá trình tải lên một tệp từ máy cá nhân lên server.
4	History	Đào Xuân Thủy	Quản lý lịch sử chat sẽ thực hiện các chức năng: thêm một Message vào file XML dùng để lưu trữ và diễn dữ liệu vào giao diện hiển thị lịch sử.
5	SocketClient	Đào Xuân Thủy	<p>Lấy những thuộc tính từ giao diện chat như port, Socket, địa chỉ máy chủ, History và các thuộc tính khác: FileOutputStream, InputStream lấy từ socket.</p> <p>Quản lý các chức năng của giao diện chat: gửi tin nhắn, nhận tin nhắn, đăng nhập, kết nối tới server, đăng ký, đăng xuất, download/upload file.</p>

Bảng 2.8 Danh mục các lớp được sử dụng trong chương trình cho Client

#### 2.2.2.2 Mô tả các phương thức của một lớp cho Client

TT	Tên phương thức	Tên các sinh viên phụ trách viết	Mục đích chính của phương thức	Input	Output	Mã giả	Tên file có sử dụng phương thức



1	<b>toString</b> (kế thừa từ interface <b>Serializable</b> )	Đào Xuân Thủy	Trả về một chuỗi kiểu String chứa các thông tin của một tin nhắn, bao gồm: người gửi, người nhận, nội dung và thời gian gửi/nhận.		String	Trả về chuỗi : {Type: loại tin nhắn, Sender: người gửi, Content: nội dung, Recipient: người nhận}	SocketClient.java: dòng 41, 187
---	---	---------------	---	--	--------	--	---------------------------------

Bảng 2.9 Bảng mô tả các phương thức trong lớp Message cho Client

T	Tên phương thức	Tên các sinh viên phụ trách viết	Mục đích chính của phương thức	Input	Output	Mã giả	Tên file có sử dụng phương thức
1	<b>Download</b> (hàm khởi tạo)	Đào Xuân Thủy	Khởi tạo một đối tượng	- String saveTo		Input: (saveTo, ui)	SocketClient : dòng 132

			Download , lấy về đường dẫn lưu file tải về và các thuộc tính từ Giao diện chat.	(địa chỉ lưu tệp) - ChatFrame (Giao diện người dùng)		Server = new ServerSocket (port 0);  Port= server.getPort ;  this.ui = ui;	
2	<b>Run</b> (kế thừa từ interface <b>Runnable</b> )	Đào Xuân Thủy	Thực hiện lưu tệp tải từ màn hình chat về máy.			Get InputStream, OutputStream (địa chỉ lưu file);  Tạo một buffer kiểu byte[1024];  Đọc dữ liệu từ buffer và dùng OutputStream để ghi;  Đóng kết nối Input/Output Stream và socket;	Vì là hàm kế thừa từ interface Runnable nên được tạo và kích hoạt chạy bên trong luồng chính và sẽ chạy song song với luồng chính.

Bảng 2.10 Bảng mô tả các phương thức trong lớp Download cho Client

<b>T T</b>	<b>Tên phương thức</b>	<b>Tên các sinh viên phụ trác h viết</b>	<b>Mục đích chính của phương thức</b>	<b>Input</b>	<b>Output</b>	<b>Mã giả</b>	<b>Tên file có sử dụng phương thức</b>
1	<b>Upload</b> (hàm khởi tạo)	Đào Xuân Thủy	Khởi tạo một đối tượng Upload, lấy về File, giao diện (dùng để thay đổi giao diện), tạo một socket bởi địa chỉ máy chủ, và cổng port, Input/Output Stream.	- String addr (địa chỉ máy chủ) - int port: cổng - File filepath: URL của file trên server - ChatFrame (Giao diện người dùng)		Input: (addr, port, file, frame)  Socket = new Socket(addr, port);  Output = socket.getOut t;  Input = new FileInput(file );	- SocketClient: dòng 152

2	<b>run</b> (kế thừa từ interface <b>Runnable</b> )	Đào Xuân Thủy	Thực hiện tải lên server từ máy khách hàng.			Tạo một buffer kiểu byte[1024]; Đọc dữ liệu từ buffer và dùng OutputStream để ghi; Thông báo upload thành công trên TextField; Đóng kết nối Input/Output Stream và socket;	Vì là hàm kế thừa từ interface Runnable nên được tạo và kích hoạt chạy bên trong luồng chính và sẽ chạy song song với luồng chính.
---	--	---------------	---	--	--	---	--

Bảng 2.11 Bảng mô tả các phương thức trong lớp Upload cho Client

T T	Tên phương thức	Tên các sinh viên phụ trách	Mục đích chính của phương thức	Input	Output	Mã giả	Tên file có sử dụng phương thức

		ch viết					
1	<b>History</b> (hàm khởi tạo)	Đào Xuân Thủy	Khởi tạo một đối tượng History, lấy về địa chỉ lưu file History.x ml (lưu trữ lịch sử chat)	- String filePath : Địa chỉ lưu file History.x ml		Input: (filePath)  Gán filePath cho thuộc tính filePath của đối tượng History.	-  ChatFrame: dòng 50, 387
2	<b>addMessage</b>	Đào Xuân Thủy	Thêm một Message vào file History.x ml để lưu trữ tin nhắn	- Message msg: một đối tượng kiểu Message đã được định nghĩa ở phần trước.  - String time: thời gian gửi tin nhắn.		- Tạo mới DocumentBuilderFactory, DocumentBuilder, Document (prase.(filePath))  - Tạo một Node data lưu giữ liệu của tin nhắn  - Tạo Element message(Element message, Element sender, Element content, Element	-  SocketClient: dòng 54, 192

						<p>recipient, Element time)</p> <ul style="list-style-type: none"> <li>- Nối Element message vào Node data</li> <li>- Tạo mới TransformerFactory, Transformer, DOMSource, StreamResult dùng để lưu lại những thay đổi vào file *.xml (file lưu lịch sử chat).</li> </ul>	
3	FillTable	Đào Xuân Thủy	Điền dữ liệu vào bảng hiển thị trên giao diện HistoryFrame	- HistoryFrame: Giao diện của màn hình xem lịch sử chat		<ul style="list-style-type: none"> <li>- Tạo một DefaultTableModel lấy từ bảng hiển thị trên giao diện xem lịch sử.</li> <li>- Tạo File fXmlFile lấy file từ thuộc tính filePath</li> <li>- Tạo mới DocumentBuilderFactory, DocumentBuilder,</li> </ul>	- HistoryFrame: dòng 20

						Document (prase.(fXmlFile))  - Tạo mộ NodeList lấy Element có tên message từ Document  - Dùng vòng lặp đọc các Node trong NodeList và thêm thông tin các tin nhắn đó vào DefaultTableModel	
4	getTagValue	Đào Xuân Thủy	Lấy giá trị của Tag theo tên Tag	- String sTag: tên Tag - Element eElement: element cần lấy giá trị theo tên	String: giá trị của Tag	- Tạo NodeList lấy tất cả các Element con theo tên Tag - Tạo Node lấy giá trị đầu tiên của NodeList - Trả về giá trị của Node	- History: dòng 75, 76

Bảng 2.12 Bảng mô tả các phương thức trong lớp History cho Client

T T	Tên phương thức	Tên các sinh viên	Mục đích chính của phương thức	Input	Output	Mã giả	Tên file có sử dụng
--------	-----------------	-------------------	--------------------------------	-------	--------	--------	---------------------

		phụ trác h viết					phương thức
1	<b>SocketClient</b> (hàm khởi tạo)	Đào Xuân Thủy	Khởi tạo một đối tượng SocketClient, lấy về giao diện của chương trình (địa chỉ máy chủ, cổng kết nối, History) đồng thời từ đó tạo ra Socket và ObjectInputStream, ObjectOutputStream	- ChatFrame (Giao diện người dùng)		Input: (ui) this.ui = ui; lấy địa chỉ máy chủ, port từ ui; - Tạo Socket từ địa chỉ máy chủ và port - Lấy Input/Output Stream từ Socket - lấy về History từ ui	- ChatFrame: dòng 302
2	<b>run</b> (kế thừa từ interface <b>Runnable</b> )	Đào Xuân Thủy	Bao gồm các chức năng: Nhận tin nhắn, đăng nhập, kết nối tới server, đăng ký, đăng xuất, upload và download file.			- In ra nội dung tin nhắn trong màn hình console: message.ToString() - try: + <b>if(type = message)</b> In ra	Vì là hàm kế thừa từ interface Runnable nên được tạo và kích hoạt chạy bên



						<p>màn hình hiển thị tin nhắn gửi đến (sender &gt; Me    sender &gt; recipient), lưu tin nhắn vào History.</p> <pre> + if(type = login) {   if(content = TRUE) Enable   các button và textfield trong   giao diện, Thông báo   “Login successfull”;   else Thông báo   lỗi “Login failed” }  + if (type = newuser) kiểm   tra username   đã tồn tại hay   chưa, nếu chưa   thì add new   user vào file   Data.</pre>	<p>trong luồng chính và sẽ chạy song song với luồng chính.</p>
--	--	--	--	--	--	--	--

						<pre> +  if(type  = signup)    { if(content  = TRUE) Enable các    button, thông    báo “Signup success”  else Thông    báo “Signup Failed” }  +  if (type  = signout) In ra cửa sổ chat của những    người đang    online “sender &gt; Me: Bye” và stop luồng    đang chạy cho người dùng hiện tại.  +  if (type  = upload_req)  Tạo popup hỏi người dùng có muốn nhận file từ người gửi hay không, nếu </pre>	
--	--	--	--	--	--	---	--

						<p>có cho người dùng chọn nơi lưu đường dẫn, tạo mới thuộc tính Download và</p> <pre> send("upload_res", "username", "Download.port", sender), nếu không send("upload_res", ui.username, "NO",sender)  + if (type = upload_res) { if(content != NO)Tạo mới một đối tượng Upload và Thread, sau đó start thread vừa tạo else In thông báo "Người nhận </pre>	
--	--	--	--	--	--	---	--

						<p>không đồng ý nhận file” }</p> <p>+ Nếu không thuộc tất cả bên trên: In ra màn hình “Không thể hiểu loại tin nhắn gửi đi”</p> <p>- catch:</p> <p>+ Enable = false các button và textfield trên giao diện</p> <p>+ Xóa tất cả nội dung trên bảng hiển thị tin nhắn</p> <p>+ dừng Thread mà người dùng đang chạy</p> <p>+ in ra thông báo: “Exception SocketClient run()”</p>	
3	<b>send</b>	Đào Xuân	Chức năng gửi tin nhắn tới một	- Messag		- In ra nội dung tin nhắn trong	- ChatFra

		n Thủ y	user khác hoặc chat tới tất cả mọi người online.	e (Đối tượng tin nhắn)		màn hình console - Nếu tin nhắn có loại “message” và nội dung khác “.bye” thì thêm tin nhắn vào History và thêm nội dung tin nhắn vào bảng hiển thị của người dùng.	me: dòng 42, 305, 319, 329, 339, 367
--	--	---------------	---	---------------------------------	--	--	---

Bảng 2.13 Bảng mô tả các phương thức trong lớp SocketClient cho Client

### 3. Phân công công việc

Sinh viên thực hiện	Đánh giá chung phần trăm đóng góp	Mô tả khái quát mảng công việc SV thực hiện trong đề án
Đào Xuân Thủy	50%	Thực hiện thiết kế và viết chương trình chat cho Client
Ngô Công An	50%	Thực hiện thiết kế và viết chương trình chat cho Server

Bảng 3.1 Mô tả phân công công việc

# KẾT LUẬN

Như vậy, đồ án hiện tại đã hoàn thành được được 80% mục tiêu đề ra.

Ưu điểm:

- Tạo ra được ứng dụng thiết thực, thân thiện, có thể tạo mọi server với địa chỉ IP riêng dùng để chat bằng nhiều máy với nhau.

Khuyết điểm:

- Chỉ thực hiện chat được với những người đang online trên Server.
- Chưa thực hiện được chứng năng tạo nhóm chat và chat nhóm, chỉ chat riêng với 1 người hoặc gửi đi tất cả mọi người.

Khó khăn:

- Chưa có hiểu biết về SocketServer, cách thức truyền tin lên server và server phản hồi lại. Từ đó bắt đầu tìm hiểu về công nghệ này và đưa ra hướng đi cho đồ án.
- Cách thức đọc/ghi file \*.xml để lưu trữ dữ liệu. Sau đó đã tham khảo trên internet và dùng cho đồ án của mình.

Hướng đi của đồ án:

- Phát triển chức năng tạo nhóm chat riêng, chat nhóm.
- Phát triển cơ sở dữ liệu riêng dùng để quản lý người dùng, lưu trữ dữ liệu chat.
- Phát triển chứng năng người dùng có thể chat tới người nhận ngay cả khi người nhận đang offline.
- Tiếp tục phát triển chức năng gửi/nhận file để có thể tải được file có dung lượng lớn hơn.

## TÀI LIỆU THAM KHẢO

1. Lê Chí Huy, *Hướng dẫn lập trình Java Socket*, Website học lập trình trực tuyến, <https://o7planning.org/vi/10393/huong-dan-lap-trinh-java-socket>, 09/05/2016.
2. Nilesh Jadav, *How To Make A Chat Application Using Sockets In Java*, Website C# Corner, <https://www.c-sharpcorner.com/article/how-to-make-a-chat-application-using-sockets-in-java/>, 17/04/2017.
3. Nguyễn Khánh, *Đọc tập tin XML sử dụng DOM*, Website hướng dẫn java trực tuyến, <https://huongdanjava.com/vi/doc-tap-tin-xml-su-dung-dom.html>, 01/07/2016.