3.1. Các biểu mẫu dùng cho thiết kế giao diện

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Màn hình/Cửa sổ/Dialog (Tên màn hình trên 1 dòng; dưới dòng này là hinh chụp màn hình) | Người thiết kế & giải thích ngắn gọn | Mục đích chính của màn hình |
| 1 | Server Screen | Ngô Công An  - Button chọn đường dẫn tới file chứa dữ liệu người dùng (\*.xml) bao gồm tài khoản và mật khẩu người dùng  - Button chạy Server  - Panel thể hiện các trạng thái của chương trình (trạng thái kết nối, client đăng nhập, …). | Tạo một giao diện đơn giản cho người quản lý server và thể hiện được các trạng thái của server khi đang chạy. |
| 2 | Client Chat Screen | Đào Xuân Thủy  - Cho phép người dùng nhập địa chỉ máy chủ, cổng kết nối và có thể kết nối tới Server.  - Đăng nhập bằng tài khoản cá nhân hoặc đăng ký tài khoản.  - Chọn file dùng để lưu lại lịch sử chat của mình.  - Button xem lại lịch sử chat.  - Panel giao diện chat.  - Cửa sổ chọn người nhận (chat với tất cả hoặc một cá nhân).  - Khung nhập tin nhắn để gửi đi.  - Chọn file để gửi file cho người nhận. | 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. |
| 3 | Chat History Screen | Đào Xuân Thủy  - Tạo một khung hiển thị nội dung của lịc 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

3.2. Các mẫu dùng cho đặc tả phần mềm (bài toán)

3.3. Các mẫu dùng cho thiết kế code Client

Bảng danh mục các lớp được sử dụng trong chương trình

|  |  |  |  |
| --- | --- | --- | --- |
| 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 điễn dữ liệu vào giao diện hiển thị lịch sử. |
| 5 | SocketClient | Đào Xuân Thủy | 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.  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 |

Mỗi lớp, mô tả các phương thức (được sử dụng trong chương trình) của 1 lớp

Mẫu bảng mô tả các phương thức trong lớp Message

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết (Nếu là code lấy từ ngồn khác thì phải chỉ rõ lấy từ nguồn nào) | Mục đích chính của phương thức trong chương trình | Input | Output | Mã giả (lưu ý viết dạng mã giả thuật toán dùng trong phương thức) | Tên file có gọi sử dụng phương thức. Liệt kê số dòng vị trí gọi. |
| 1 | toString (kế thừa từ interface **Serializable** ) | 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 |

Mẫu bảng mô tả các phương thức trong lớp Download

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết (Nếu là code lấy từ ngồn khác thì phải chỉ rõ lấy từ nguồn nào) | Mục đích chính của phương thức trong chương trình | Input | Output | Mã giả (lưu ý viết dạng mã giả thuật toán dùng trong phương thức) | Tên file có gọi sử dụng phương thức. Liệt kê số dòng vị trí gọi. |
| 1 | Download (hàm khởi tạo) | Đào Xuân Thủy | Khởi tạo một đối tượng Download, lấy về đường dẫn lưu file tải về và các thuộc tính từ Giao diện chat. | - String saveTo (địa chỉ lưu tệp)  - ChatFrame (Giao diện người dùng) |  | Input: (saveTo, ui)  Server = new ServerSocket (port 0);  Port = server.getPort;  this.ui = ui; | - SocketClient: dòng 132 |
| 2 | run (kế thừa từ interface **Runnable** ) | 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. |

Mẫu bảng mô tả các phương thức trong lớp Upload:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết (Nếu là code lấy từ ngồn khác thì phải chỉ rõ lấy từ nguồn nào) | Mục đích chính của phương thức trong chương trình | Input | Output | Mã giả (lưu ý viết dạng mã giả thuật toán dùng trong phương thức) | Tên file có gọi sử dụng phương thức. Liệt kê số dòng vị trí gọi. |
| 1 | Upload (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;  Input = new FileInput(file); | - SocketClient: dòng 152 |
| 2 | run (kế thừa từ interface **Runnable** ) | Đào Xuân Thủy | Thực hiện tải lên server tệp 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. |

Mẫu bảng mô tả các phương thức trong lớp History:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết (Nếu là code lấy từ ngồn khác thì phải chỉ rõ lấy từ nguồn nào) | Mục đích chính của phương thức trong chương trình | Input | Output | Mã giả (lưu ý viết dạng mã giả thuật toán dùng trong phương thức) | Tên file có gọi sử dụng phương thức. Liệt kê số dòng vị trí gọi. |
| 1 | History (hàm khởi tạo) | Đào Xuân Thủy | Khởi tạo một đối tượng History, lấy về địa chỉ lưu file History.xml (lưu trữ lịch sử chat) | - String filePath :  Địa chỉ lưu file History.xml |  | Input: (filePath)  Gán filePath cho thuộc tính filePath của đối tượng History. | - ChatFrame: dòng 50, 387 |
| 2 | addMessage | Đào Xuân Thủy | Thêm một Message vào file History.xml 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 gữ liệu của tin nhắn  - Tạo Element message(Element message, Element sender, Element content, Element recipient, Element time)  - Nối Element message vào Node data  - Tạo mới TransformerFactory, Transformer, DOMSource, StreamResult dùng để lưu lại những thay đổi vào file \*.xml (file lưu lịch sử chat). | - SocketClient: dòng 54, 192 |
| 3 | FillTable | Đào Xuân Thủy | Điền dữ liệu vào bảng hiển thị trên giao diện HistoryFrame | - HistoryFrame frame: Giao diện của màn hình xem lịch sử chat |  | - Tạo một DefaultTableModel lấy từ bảng hiển thị trên giao diện xem lịch sử.  - Tạo File fXmlFile lấy file từ thuộc tính filePath  - Tạo mới DocumentBuilderFactory, DocumentBuilder, 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 | - HistoryFrame: dòng 20 |
| 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 cac Element con theo tên Tag  - Tạo Node lấy giá trị đầu tiên của NodeList  - Trả về giá trị của Node | - History: dòng 75, 76 |

Mẫu bảng mô tả các phương thức trong lớp SocketClient

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TT | Tên phương thức | Tên các SV phụ trách viết (Nếu là code lấy từ ngồn khác thì phải chỉ rõ lấy từ nguồn nào) | Mục đích chính của phương thức trong chương trình | Input | Output | Mã giả (lưu ý viết dạng mã giả thuật toán dùng trong phương thức) | Tên file có gọi sử dụng phương thức. Liệt kê số dòng vị trí gọi. |
| 1 | **SocketClient** (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 | **run** (kế thừa từ interface **Runnable** ) | 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:  + **if**(**type** = **message**) In ra màn hình hiển thị tin nhắn gửi đến (sender > Me || sender > recipient), lưu tin nhắn vào History.  + **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.  + if(type = signup) { if(content = TRUE) Enable các button, thông báo “Signup success” else Thông báo “Signup Failded” }  + if (type = signout) In ra cửa sổ chat của những người đang online “sender > 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 có cho người dùng chọn nơi lưu đường dẫn, tạo mới thuộc tính Download và 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 không đồng ý nhận file” }  + Nếu không thuộc tất cả bên trên: In ra màn hình “Không thể hiểu loại tin nhắn gửi đi”  - catch:  + Enable = false các button và textfield trên giao diện  + Xóa tất cả nội dung trên bảng hiển thị tin nhắn  + dừng Thread mà người dùng đang chạy  + in ra thông báo: “Exception SocketClient run()” | 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. |
| 3 | **send** | Đào Xuân Thủy | Chức năng gửi tin nhắn tới một user khác hoặc chat tới tất cả mọi người online. | - Message (Đối tượng tin nhắn) |  | - In ra nội dung tin nhắn trong màn hình console  - Nếu tin nhắn có loại “mesage” và nội dung khác “.bye” thì thêm tin nhắn vào History và thêm nội dung tin nhắn vào bảng hiển thị của người dùng. | - ChatFrame: dòng 42, 305, 319, 329, 339, 367 |