**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM**



**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**🕯✡🕮🕮✡🕯**

****

**BÁO CÁO ĐỒ ÁN I**

***ĐỀ TÀI:* VIẾT CHƯƠNG TRÌNH REMOTE DRAW TẠO MÔI TRƯỜNG LÀM VIỆC CỘNG TÁC CHO PHÉP NHIỀU USERS CÙNG VẼ TRÊN MỘT BẢN VẼ Ở XA**

**GVHD**: ThS. Nguyễn Đăng Quang

**Sinh viên thực hiện**:

Trần Minh Hiếu 17110135

Nguyễn Ngọc Tú Duyên 17110116

Võ Thị Vân Anh 17110098

***TP. Hồ Chí Minh, ngày tháng 5 năm 2019***

NHẬN XÉT CỦA GIẢNG VIÊN

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....

……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………....

MỤC LỤC

[NHẬN XÉT CỦA GIẢNG VIÊN 2](#_Toc24298655)

[MỤC LỤC 3](#_Toc24298656)

[LỜI NÓI ĐẦU 1](#_Toc24298657)

[PHẦN MỞ ĐẦU 2](#_Toc24298658)

[**I.** **Lý do chọn đề tài:** 2](#_Toc24298659)

[**II.** **Mục tiêu cho đề tài:** 2](#_Toc24298660)

[**III.** **Phương pháp nghiên cứu:** 2](#_Toc24298661)

[NỘI DUNG 3](#_Toc24298662)

[**I.** **ĐẶC TẢ.** 3](#_Toc24298663)

[**1.** **Giới thiệu chung về đề tài.** 3](#_Toc24298664)

[**1.1.** **Giới thiệu đề tài.** 3](#_Toc24298665)

[**1.2.** **Mô hình đối tượng.** 3](#_Toc24298666)

[**1.3.** **Các chức năng của ứng dụng.** 3](#_Toc24298667)

[**2.** **Tìm hiểu ngôn ngữ Java và môi trường Java Swing.** 4](#_Toc24298668)

[**2.1.** **Ngôn ngữ Java.** 4](#_Toc24298669)

[**2.2.** **Môi trường Java Swing.** 4](#_Toc24298670)

[**2.3.** **Tìm hiểu SocketIO.** 4](#_Toc24298671)

[**II.** **PHÂN CÔNG CÔNG VIỆC.** 5](#_Toc24298672)

[**III.** **THIẾT KẾ.** 6](#_Toc24298673)

[**1.** **Thiết kế sơ đồ UseCase.** 6](#_Toc24298674)

[**1.1.** **Mô tả chi tiết UseCase “Login”.** 6](#_Toc24298675)

[**1.2.** **Mô tả chi tiết UseCase “Manipulate Drawing Tools”.** 7](#_Toc24298676)

[**1.3.** **Mô tả chi tiết UseCase “Manipulate Workspace”.** 7](#_Toc24298677)

[**1.4.** **Mô tả chi tiết UseCase “User Management”.** 8](#_Toc24298678)

[**2.** **Thiết kế sơ đồ Class Diagram.** 8](#_Toc24298679)

[**2.1.** **Sơ đồ tổng thể ứng dụng.** 8](#_Toc24298680)

[**2.2.** **Sơ đồ thiết kế Shape.** 9](#_Toc24298681)

[**2.3.** **Sơ đồ thiết kế SaveFile** 10](#_Toc24298682)

[**2.4.** **Sơ đồ thiết kế WhiteboardClient.** 10](#_Toc24298683)

[**2.5.** **Sơ đồ thiết kế WhiteboardServer.** 11](#_Toc24298684)

[**2.6.** **Sơ đồ thiết kế SocketClient.** 11](#_Toc24298685)

[**2.7.** **Sơ đồ thiết kế SocketServer.** 12](#_Toc24298686)

[**3.** **Thiết kế lớp.** 12](#_Toc24298687)

[**3.1.** **Bảng danh mục các lớp sử dụng trong chương trình.** 13](#_Toc24298688)

[**3.1.1.** **Paint(Shape).** 13](#_Toc24298689)

[**3.1.2.** **Object.** 14](#_Toc24298690)

[**3.1.3.** **PaintApp.** 16](#_Toc24298691)

[**3.1.4.** **Frame.** 16](#_Toc24298692)

[**3.1.5.** **Socket.** 18](#_Toc24298693)

[**3.2.** **Bảng mô tả phương thức trong một lớp** 18](#_Toc24298694)

[**3.2.1.** **Class WhiteboardClient** 18](#_Toc24298695)

[**3.2.2.** **Class WhiteboardServer** 19](#_Toc24298696)

[**3.2.3.** **Class Client** 19](#_Toc24298697)

[**3.2.4.** **Class Server** 20](#_Toc24298698)

[**4.** **Thiết kế giao diện.** 21](#_Toc24298699)

[**IV.** **CÀI ĐẶT VÀ KIỂM THỬ.** 24](#_Toc24298700)

[**1.** **Cài đặt.** 24](#_Toc24298701)

[**2.** **Kiểm thử.** 24](#_Toc24298702)

[**V.** **KẾT LUẬN** 26](#_Toc24298703)

[**1.** **Kết quả đạt được:** 26](#_Toc24298704)

[**2.** **Hạn chế:** 26](#_Toc24298705)

[**3.** **Hướng phát triền:** 26](#_Toc24298706)

[**VI.** **TÀI LIỆU THAM KHẢO** 27](#_Toc24298707)

[**1.** **Tài liệu tham khảo phần Paint:** 27](#_Toc24298708)

[**2.** **Tài liệu tham khảo phần Socket:** 27](#_Toc24298709)

# **LỜI NÓI ĐẦU**

Ngày nay, ngành Công nghệ thông tin phát triển với tốc độ chóng mặt, đặc biệt là internet ngày càng phổ biến. Nó thúc đẩy quá trình phẳng hóa thế giới nhanh hơn.Nhu cầu học tập trực tuyến ngày càng cao.Từ đó, các công nghệ hỗ trợ học tập dần đa dạng hơn ví dụ : Các lớp học online…Chúng ta có thể ngồi tại nhà và tham gia các khóa học một cách chủ động hơn. Ứng dụng Remote Paint cũng dựa trên những ưu điểm đó, mô phỏng một cách trực quan về việc học tập và đóng góp xây dựng ý kiến trực tuyến. Giúp cho việc học trở nên hiện đại, dễ dàng, nhanh chóng và bắt kịp với nền công kịp mới, nền công nghiệp 4.0.

Vì vậy chúng em xin đóng góp và phát triển ý tưởng xây dựng ứng dụng Remote Draw tạo môi trường làm việc cộng tác cho phép nhiều Users cùng vẽ trên một bản vẽ ở xa.

# **PHẦN MỞ ĐẦU**

1. **Lý do chọn đề tài:**

Dựa trên những đề tài được thầy gợi ý, nhóm em nhận thấy đề tài :” Viết chương trình Remote Draw tạo môi trường làm việc cộng tác cho phép nhiều Users cùng vẽ trên một bản vẽ ở xa” rất thú vị và ứng dụng vào thực tế rất tốt, dựa trên những kiến thức đã được học về môn Lập trình Hướng đối tượng và một số kiến thức tự tìm hiểu, nhóm em mong muốn có thể nhận và hoàn thành đề tài này theo sự hướng dẫn của thầy.

1. **Mục tiêu cho đề tài:**

Vì là chương trình vẽ nên đầu tiên mục tiêu cần đạt được là 1 ứng dụng vẽ cơ bản với những công cụ thiết yếu. Tiếp theo, chúng em muốn phát triển đề tài theo hướng một môi trường vẽ đa người sử dụng.

1. **Phương pháp nghiên cứu:**

Ngoài những kiến thức được học từ môn Lập trình Hướng đối tượng, chúng em tìm kiếm thêm những tài liệu về Java và SocketIO, cũng như những đồ án mẫu trên Internet để tham khảo. Những vấn đề chưa hiểu và chưa đạt, chúng em cố gắng xin ý kiến từ thầy để hoàn thành bài một cách tốt hơn.

# **NỘI DUNG**

1. **ĐẶC TẢ.**
2. **Giới thiệu chung về đề tài.**
   1. **Giới thiệu đề tài.**

* *Tên đề tài:* Viết chương trình Remote Draw tạo môi trường làm việc cộng tác cho phép nhiều Users cùng vẽ trên một bản vẽ ở xa.
* *Mục đích:* Áp dụng những kiến thức về lập trình trên môi trường Java Swing bằng ngôn ngữ Java và những kiến thức liên quan để xây dựng một ứng dụng hoàn chỉnh.
* *Yêu cầu:* Chương trình phải đáp ứng đủ các chức năng vẽ trên máy tính và nhiều Users(Client) phải kết nối được với Server.
* *Quy trình làm việc:*
* Thống nhất đề tài đồ án.
* Phân công nhiệm vụ cho từng thành viên.
* Tìm hiểu tài liệu liên quan đến việc thực hiện đồ án.
* Lên kế hoạch các mốc thời gian cho đồ án
* Tiến hành các buổi họp nhóm để trao đổi thông tin, bàn bạc kế hoạch nội dung thực hiện, phân công nhiệm vụ và giao nhiệm vụ deadline.
* Hoàn thành đồ án, viết báo cáo và ra sản phẩm hoàn chỉnh.
* *Công cụ:* Eclipe
  1. **Mô hình đối tượng.**
* Tạo 1 class là danh sách đối tượng với kiểu List dùng để lưu lại tất cả các đối tượng được vẽ trên trang vẽ để thuận tiện cho việc chỉnh sửa đối tượng về sau.
* Mỗi đối tượng hình vẽ gồm:
* Thuộc tính: màu bút vẽ, độ đậm bút vẽ, điểm bắt đầu, điểm kết thúc, khu vực đối tượng chiếm giữ, loại hình.
* Phương thức: khởi tạo, vẽ, di chuyển, thay đổi kích thước, kiểm tra vị trí của điểm hiện hành so với đối tượng.
* Cách thức thực hiện: xây dựng các lớp đối tượng kế thừa nhau, thuận tiện cho việc duy trì, lưu trữ, chỉnh sửa.
  1. **Các chức năng của ứng dụng.**

1. *Chức năng vẽ:* sử dụng các lệnh có sẵn để vẽ các đối tượng hình cơ bản như đường thẳng, hình chữ nhật, hình vuông, hình tròn, hình tam giác, hình elip,…
2. Chức năng chọn màu của bút vẽ.
3. Chức năng mở một file đã lưu về.
4. Chức năng save: Lưu hình ảnh dưới dạng file .png hoặc .txt
5. Chức năng xóa: Xét vị trí tương đối của tọa độ chuột hiện hành và đối tượng hình có cho phép bất chức năng xóa không.
6. Chức năng chấp nhận phiên hoạt động của user khi đăng nhập vào hệ thống.
7. Chức năng xóa người dùng phiên hoạt động của user.
8. Chức năng nhắn tin giữa các Client.
9. **Tìm hiểu ngôn ngữ Java và môi trường Java Swing.**
   * 1. **Ngôn ngữ Java.**
     2. **Môi trường Java Swing.**
     3. **Tìm hiểu SocketIO.**
10. **PHÂN CÔNG CÔNG VIỆC.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên công việc** | **Trần Minh Hiếu** | **Nguyễn Ngọc Tú Duyên** | **Võ Thị Vân Anh** |
| Tìm hiểu về ngôn ngữ java và OOP trong java. | X | X | X |
| Phân tích đề tài, thiết kế lớp, đối tượng. | X | X | X |
| Thiết kế UseCase | X | X | X |
| Thiết kế giao diện | X | X | X |
| Code phần Paint | X |  | X |
| Code phần Socket | X | X |  |
| Cài đặt và kiểm thử | X | X | X |
| Viết báo cáo | X | X | X |
| Mức độ hoàn thành | 100% | 100% | 100% |

1. **THIẾT KẾ.**
2. **Thiết kế sơ đồ UseCase.**

Ứng dụng sẽ có 2 đối tượng sử dụng chính là User và Manager.

* User bao gồm việc login, tạo môi trường làm việc, các thao tác vẽ.
* Manager quản lý không gian vẽ và quản lý user.
  1. **Mô tả chi tiết UseCase “Login”.**



Mỗi user khi login vào sẽ nhập port và IP address.

* 1. **Mô tả chi tiết UseCase “Manipulate Drawing Tools”.**



Các thao tác hỗ trợ vẽ bao gồm: thêm nét vẽ, chỉnh sửa (di chuyển, đổ màu), xóa.

* 1. **Mô tả chi tiết UseCase “Manipulate Workspace”.**



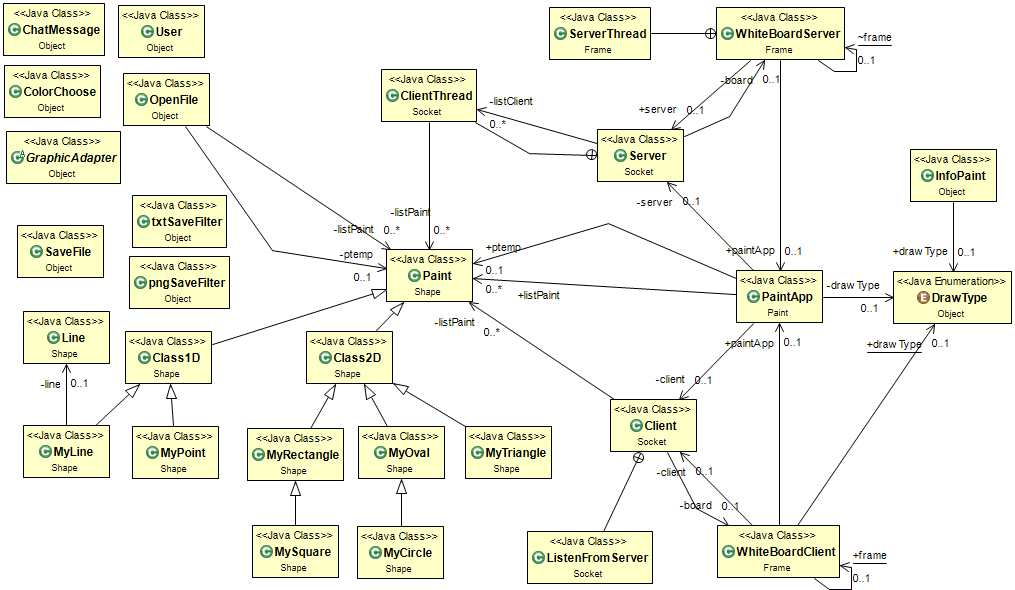
Không gian vẽ bao gồm quản lý nét vẽ của mỗi người dùng, lưu phiên làm việc, quản lý phiên làm việc.

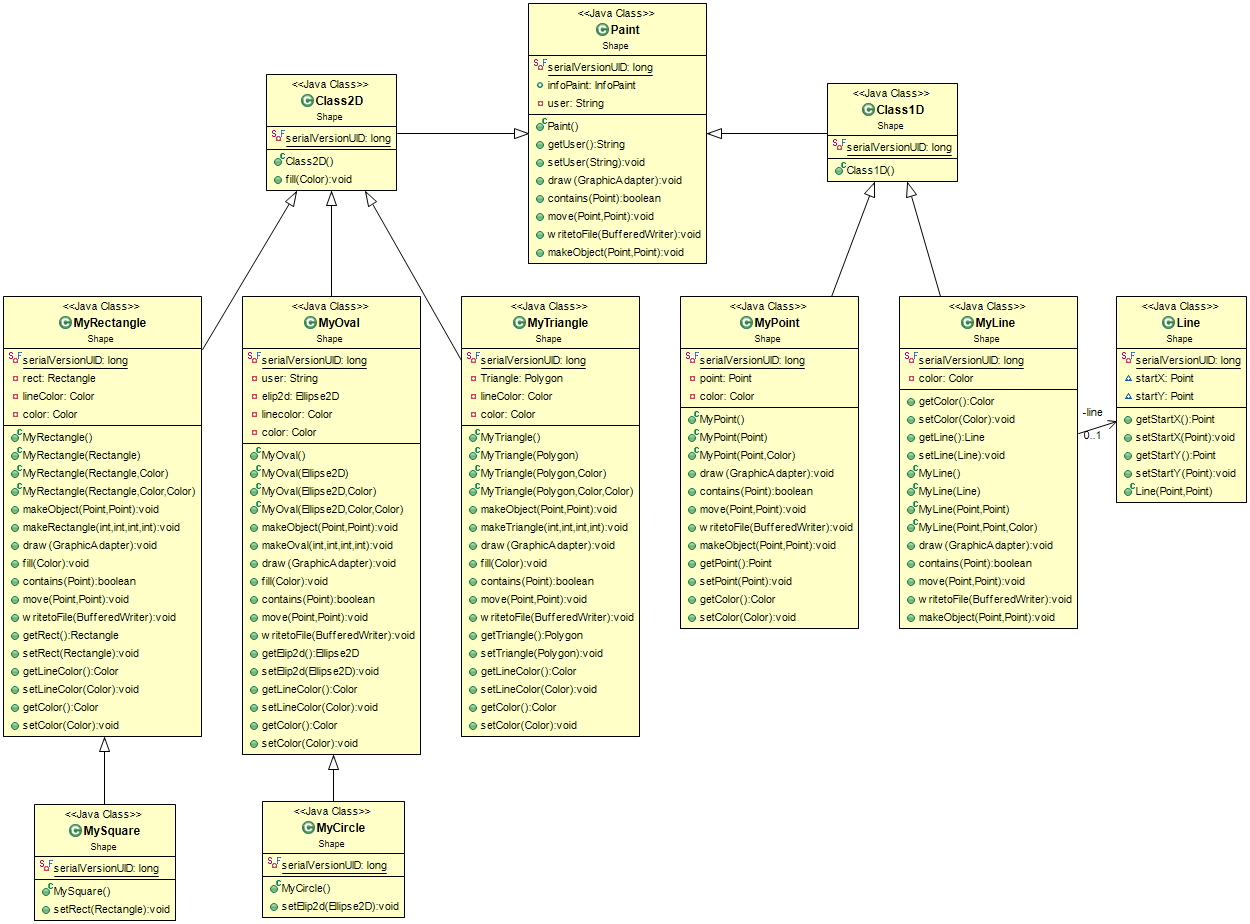
* 1. **Mô tả chi tiết UseCase “User Management”.**



Quản lý người dùng bao gồm việc quản lý quyền login vào server và quản lý số lượng user tham gia.

1. **Thiết kế sơ đồ Class Diagram.**
   1. **Sơ đồ tổng thể ứng dụng.**

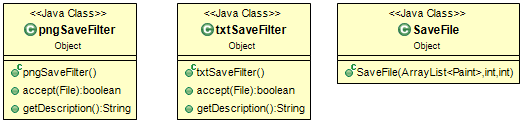
****

* 1. **Sơ đồ thiết kế Shape.**

Đầu tiên, ta sẽ thiết kế class Paint, kế đến ta tạo 2 là Class1D và Class2D kế thừa class Paint.

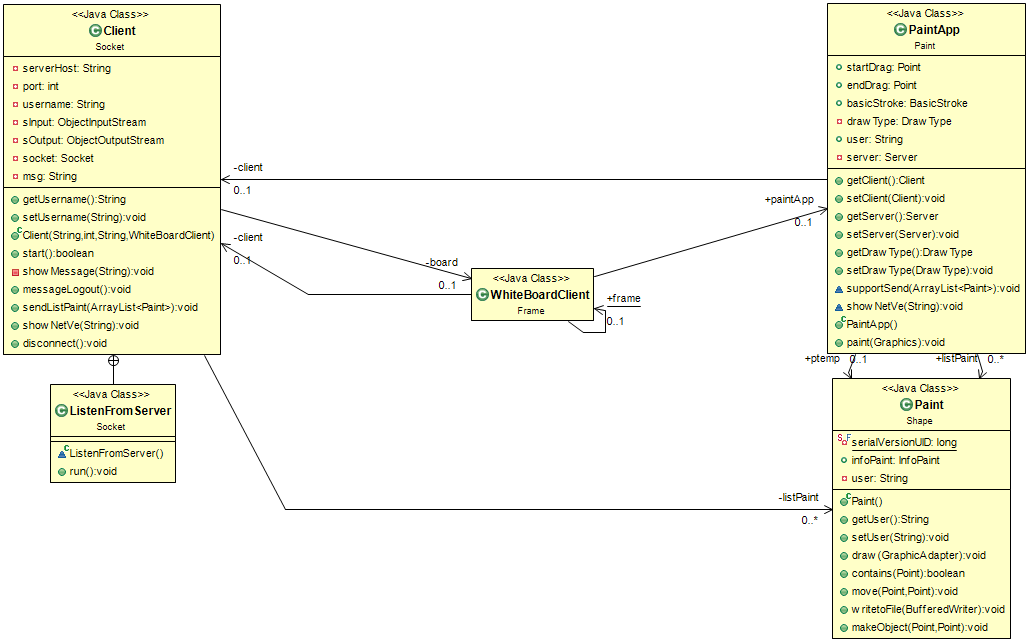
Từ Class1D ta tạo ra 2 class kế thừa đến là MyPoint và MyLine, Class Line khi dùng sẽ gọi lại class MyLine.

Từ Class2D, ta tạo ra 3 class kế thừa đến là MyRectagle, MyTriangle và MyOval, trong đó class MySquare và MyCircle lần lượt gọi lại class MyRectangle, MyOval.

* 1. **Sơ đồ thiết kế SaveFile**

Lựa chọn lưu file dưới dạng .png và .txt.

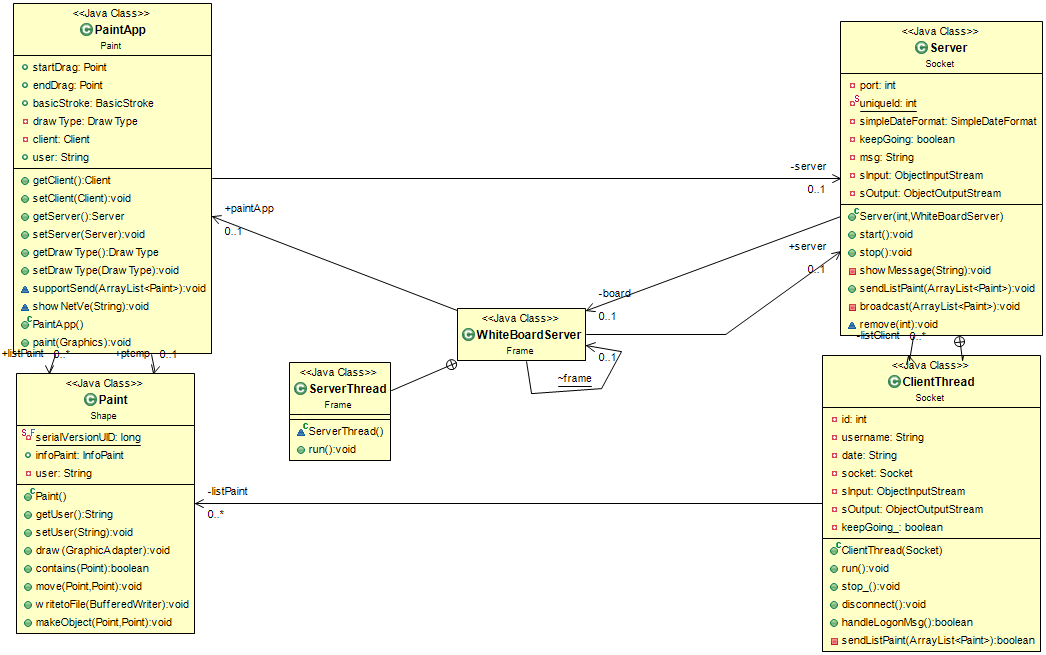
* 1. **Sơ đồ thiết kế WhiteboardClient.**



Đầu tiên, người dùng có thể mở và dùng ứng dụng offline hoặc online (thông qua socket), WhiteboardClient sẽ trả về 1 giao diện vẽ với các đối tượng để vẽ.

WhiteboardClient sử dụng PaintApp để tạo môi trường và gán các event chuột. Mỗi khi PaintApp có thao tác thì sẽ gọi lại Class Paint để thực thi.

* 1. **Sơ đồ thiết kế WhiteboardServer.**

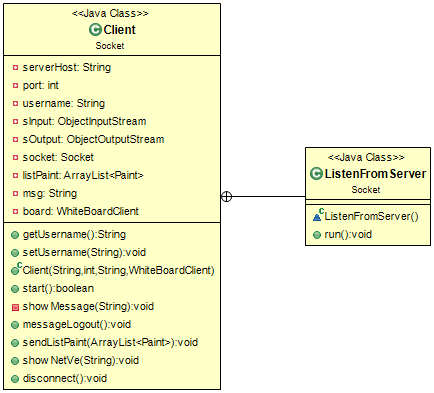


Đầu tiên, manager có thể mở và dùng ứng dụng offline hoặc online (thông qua socket), WhiteboardServer sẽ trả về 1 giao diện vẽ với các đối tượng để vẽ.

WhiteboardServer sử dụng PaintApp để tạo môi trường và gán các event chuột. Mỗi khi PaintApp có thao tác thì sẽ gọi lại Class Paint để thực thi.

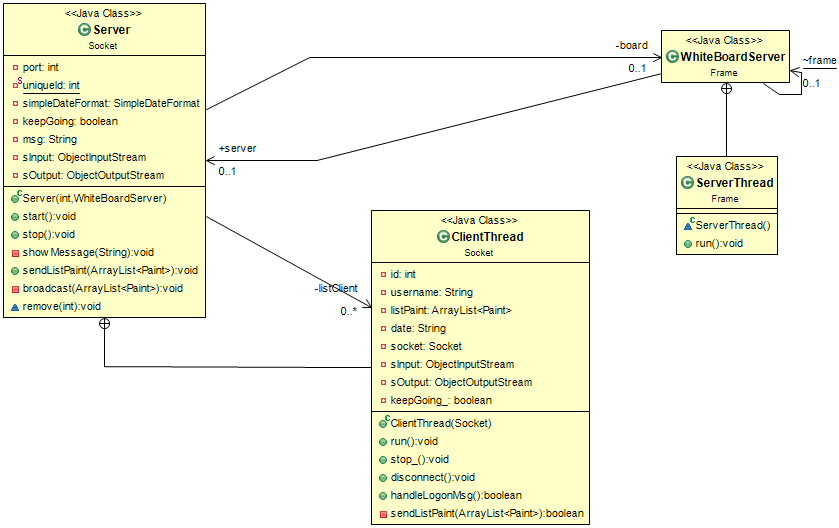
Vì Server có quyền quản lý lên user nên Server có trách nhiệm mở luồng (ServerThread) và nhận luồng (ClientThread).

* 1. **Sơ đồ thiết kế SocketClient.**

****

Chịu trách nhiệm mở luồng cho client để tham gia vào phiên hoạt động, nhận gói broadcast chứa listPaint từ Server.

* 1. **Sơ đồ thiết kế SocketServer.**

****

Mở luồng để nhận gói tin gửi đến từ client thông qua socketServer và đồng thời broadcast lại các gói tin đến tất cả các client.

1. **Thiết kế lớp.**

Như ta đã đề cập ở phần mô tả các chức năng của ứng dụng thì hai phần quan trọng nhất của ứng dụng sẽ là vẽ và sau đó sẽ là socket truyền nét vẽ đi.

Về phần vẽ: Trong phần vẽ thì ta xây dựng trước 1 lớp Paint(Shape) để mô tả các object vẽ(như là đường thẳng, vẽ tự do, các function hỗ trợ vẽ như draw, contains,…). Trong object Paint ta lại chia thành 2 lớp đó là Class1D và Class2D vì có chứ năng fill ở những hình 2D. Kế đến ta làm rõ các object(MyLine, MyCircle, MyPoint, MyOval, MyRectangle, MySquare, MyTriangle) khi kế thừa từ Class1D và Class2D. Và để có thể ứng dụng vẽ được cái object đấy lên phiên làm việc(form) thì ta cần thêm cái event hỗ trợ có trong JComponent và ta xây dựng thêm 1 lớp PaintApp để đảm nhận nhiệm vụ đó. Trên class PaintApp ta xử lý các event chuột để thao tác vẽ hình.

Về phần hỗ trợ vẽ: Xây dựng lớp ColorChoose để chọn màu cho nét vẽ, để chọn kiểu vẽ ta xây dựng thêm enum DrawType, chuyển và ép kiểu mọi dạng Graphics và Graphics2D thành Graphics2D ta xây dựng GraphicAdapter. Ngoài ra còn các hàm phụ trợ lưu và mở file.

Về phần socket: Ta đã có hai lớp là Paint(Shape) và ChatMessage được mô tả thành dạng Serialization Object để gửi. Ta xây dựng trong lớp Server các function có chứ năng mở socket, broadcast nét vẽ hoặc tin nhắn cho Client(gửi ở dạng ObjectStream), chấp nhập client, xóa client,… phục vụ cho việc quản trị viên. Lớp Client được xây dựng bên trong ruột là các function phục vụ cho việc yêu cầu kết nối tới server, logout ra khỏi server, gửi các Serialization Object(ArrayList<Paint>, ChatMessage) ở dạng ObjectStream lên server để broadcast.

Về phần form hiển thị làm việc: Gồm có hai class đó là WhiteBoardServer và WhiteBoardClient. Đã tìm hiểu Swing kết hợp với WindowBuilder nên việc thao tác cũng như thế kế cho người dùng dễ dàng sử dụng, hai form có cùng cơ chế vẽ như nhau đề dùng JComponent từ class PaintApp để vẽ chỉ khác ở chỗ mỗi form có phần socket riêng của client và server.

* 1. **Bảng danh mục các lớp sử dụng trong chương trình.**

Theo thứ tự thực hiện, nhóm sẽ thiết kế theo thứ tự sau. Đầu tiên, nhóm sẽ thiết kế Shape, Object, Paint, Frame, và cuối cùng là Socket.

* + 1. **Paint(Shape).**

Chứa các đối tượng vẽ cơ bản như điểm, đường thẳng, hình tam giác, hình vuông, hình tròn, hình oval, hình tam giác vuông.

Mô tả:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên Lớp** | **Mục đích** |
| 1 | Paint | Là class khai báo các hàm chức năng, được các class khác trong package Shape gọi lại và định nghĩa.  Draw – vẽ đối tượng  Contains- kiểm tra điểm có thuộc đối tượng không  Move- di chuyển đối tượng  Writetofile- ghi các giá trị của đối tượng ra file text  Makeobject- tạo ra đối tượng |
| 2 | Class1D | Kế thừa Paint  Bao gồm vẽ tự do và vẽ đường thẳng |
| 3 | Class MyPoint | Được kế thừa từ Class1D  Là lớp vẽ tự do |
| 4 | Class MyLine | Được kế thừa từ Class1D  Là lớp vẽ đường thẳng |
| 5 | Class2D | Bao gồm vẽ hình chữ nhật, hình vuông, hình tròn, hình oval, hình tam giác vuông.  Gọi là hàm fill() – đổ màu cho các đối tượng. |
| 6 | Class MyRectangle | Lớp vẽ hình chữ nhật |
| 7 | Class MySquare | Lớp vẽ hình vuông |
| 8 | Class MyCircle | Lớp vẽ hình tròn |
| 9 | Class MyOval | Lớp vẽ hình oval |
| 10 | Class MyTriangle | Lớp vẽ hình tam giác vuông |

* + 1. **Object.**

Đặc tả các lớp chức năng của ứng dụng.

Mô tả:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên lớp** | **Mục đích** |
| 1 | Class DrawType | Sử dụng Serializable  Liệt kê các kiểu vẽ như điểm, đường thẳng, hình chữ nhât, hình vuông, hình tròn, hình oval, hình tam giác vuông.  Một số thao tác như: di chuyển hình vẽ, xóa hình vẽ khỏi panel, open file, save file dưới dạng .png và .txt, tô màu cho các hình vẽ thuộc Class2D. |
| 2 | Class InfoPaint | Sử dụng Serializable  Lớp này chứa các thông tin về điểm vẽ (cụ thể là điểm bắt đầu và điểm kết thúc), màu vẽ (khởi tạo ban đầu là màu đen) và độ dày của nét vẽ (khởi tạo ban đầu = 1) |
| 3 | Class ColorChoose | Chọn màu vẽ |
| 4 | Class OpenFile | Lựa chọn dạng file để mở file dưới dạng .png hoặc là .txt |
| 5 | Class SaveFile | Lựa chọn lưu file dưới dạng .png và .txt |
| 6 | Class pngSaveFilter | Lưu file dưới dạng .png |
| 7 | Class txtSaveFilter | Lưu file dưới dạng .txt |
| 8 | Class User | Sử dụng Serializable  Gồm ID và tên user |
| 9 | Class ChatMessage | Sử dụng Serializable  Lớp này định nghĩa trao đổi tin nhắn giữa Client và Server.  -Login: Dùng để đăng nhập vào server  -Message: Những thông điệp dạng chuỗi được gửi lên Server  -Logout: dùng thể thoát/ngắt kết nối với server. |

* + 1. **PaintApp.**

Tạo môi trường để vẽ và gán các Event chuột

Mô tả:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên lớp** | **Mục đích** |
| 1 | Class Paint\_App | Tạo môi trường để vẽ và gán các Event chuột như là:  **public** **void** mousePressed(MouseEvent e), **public** **void** mouseReleased(MouseEvent e), **public** **void** mouseDragged(MouseEvent e) |

* + 1. **Frame.**

Gồm WhiteboardClient (dành cho users) và WhiteboardServer (dành cho managers).Ở giao diện cho users, người dùng có các công cụ để vẽ, tạo mới bản vẽ, lưu hình đã vẽ, chatroom... Ở giao diện cho manager, ta có thêm mục quản lý user đang đăng nhập trong phiên.

Mô tả:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên lớp** | **Mục đích** |
| 1 | Class WhiteboardClient | Tạo Menubar gồm các chức năng như tạo newpaint, mở bản vẽ, lưu bản vẽ (dưới dạng .png hoặc .txt) và thoát ứng dụng.  Panel paint thực hiện các thao tác vẽ đối tượng lên panel như vẽ điểm, đường thẳng, hình tròn, hình vuông, hình chữ nhật, hình oval, hình tam giác vuông với các thao tác như là đổ màu với những hình thuộc Class2D, di chuyển hình, xóa hình.  User có thể chọn lựa việc vẽ offline hoặc đăng nhập lên server(port) để login vào môi trường đa người sử dụng, các thao tác của user sẽ được real- time trên ứng dụng, các user tham gia trong phiên sẽ có thể nhìn thấy và tương tác lên các đối tượng trên panel.  Mỗi user đăng nhập vào sẽ được server xác nhận được phép đăng nhập hay không? Nếu có, user sẽ được thêm vào list user được server quản lý.  Tất cả các user đăng nhập vào hệ thống có thể gửi tin nhắn đến server. |
| 2 | Class WhiteboardServer | Tạo Menubar gồm các chức năng như tạo newpaint, mở bản vẽ, lưu bản vẽ (dưới dạng .png hoặc .txt) và thoát ứng dụng.  Panel paint thực hiện các thao tác vẽ đối tượng lên panel như vẽ điểm, đường thẳng, hình tròn, hình vuông, hình chữ nhật, hình oval, hình tam giác vuông với các thao tác như là đổ màu với những hình thuộc class2D, di chuyển hình, xóa hình.  Manager có thể sử dụng whiteboardserver để vẽ offline với đầy đủ tính năng.Manager khởi chạy Whiteboardserver để bắt đầu 1 luồng mang số port và địa chỉ máy chủ (localhost). Mỗi khi có user tương tác lên server, server sẽ nhận gói tin và gửi lại 1 gói broadcast cho tất cả các user đang tham gia vào phiên.  Khởi tạo 1 list user để manager có thể dễ dàng quản lý được phiên đăng nhập của từng user. Khi user đăng nhập vào hệ thống, manager sẽ xác nhận user được phép đăng nhập vào hệ thống hay không? Nếu có, add user vào list user.  Manager có quyền theo dõi việc đăng nhập/ xuất của user và kick một hoặc nhiều user bất kì khỏi phiên. Manager dùng double click vào username nào bất kì mà manager muốn loại bỏ khỏi list user.  Server sẽ nhận được tất cả các tin nhắn của các user gửi lên, đồng thời tin nhắn sẽ được hiển thị trong phần chatroom. |

* + 1. **Socket.**

Do là ứng dụng truyền dữ liệu realtime, ta sử dụng SocketIO giúp server và client kết nối với nhau cụ thể là mọi hình ảnh được user vẽ trên panel client đều được , gửi ngay lập tức đến server và truyền dữ liệu ngay lập tức thông qua server trung gian.

Mô tả:

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên lớp** | **Mục đích** |
| 1 | Class Server | Mỗi server đều có 1 uniqueID và 1 port, một danh sách users truy cập, thời gian truy cập của từng user. Khi manager truy cập vào app sẽ được yêu cầu nhập số port để bắt đầu server và đồng thời duy trì luồng kết nối đến khi manager disconnect khỏi server socket |
| 2 | Class Client | Mỗi user muốn tham gia vào phiên sẽ điền vào form bao gồm địa chỉ server (Server Address), số Port (Port Number) và tên người dùng. Socket của client sẽ tạo và duy trì luồng đăng nhập của client vào server. |

* 1. **Bảng mô tả phương thức trong một lớp**

1. **Class WhiteboardClient**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên hàm** | **Mục đích** |
| 1 | **public** **void** append(String str) | Dùng để thêm văn bản vào textArea  Xác định vị trí cần thêm đoạn văn bản và thêm chuỗi theo ý muốn ngay sau vị trí đã xác định trước đó. |
| 2 | **public** **void** connectionFailed() | Được gọi lên bởi WhiteboardClient khi kết nối không thành công  Ta sẽ reset các button, label, textfield tức là user sẽ điền lại thông tin của server address và port number để tham gia vào phiên hoạt động. |

1. **Class WhiteboardServer**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên hàm** | **Mục đích** |
| 1 | **public** **void** appendChatRoom(String str) | Nối tin nhắn vào JtextArea  Thêm tin nhắn theo ý muốn ngay sau vị trí đã xác định trước đó. |
| 2 | **public** **void** appendEvent(String str) | Thêm vào 1 sự kiện |

1. **Class Client**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên hàm** | **Mục đích** |
| 1 | **public** **boolean** start() | -User tạo 1 kết nối đến Server  +Nếu kết nối thành công sẽ trả về 1 địa chỉ và số port của server được kết nối đến  +Nếu kết nối thất bại sẽ báo 1 thông điệp “Error connecting to server – IOException”  -Tạo data stream  -Tạo thread để lắng nghe server  -Gửi username đến server |
| 2 | **public** **void** sendListPaint(ArrayList<Paint> listPaint) | Gửi 1 list Paint đến server |
| 3 | **private** **void** disconnect() | Khi có lỗi xảy ra, đóng input/output stream và ngắt kết nối đến server. |
| 4 | **class** ListenFromServer **extends** Thread {  **public** **void** run()  } | Lớp này sẽ chờ những thông điệp từ server  Hàm run() listPaint sẽ nhận giá trị là 1 list đối tượng và vẽ lại. |

1. **Class Server**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên hàm** | **Mục đích** |
| 1 | **public** **void** start() | Tạo socket server và chờ đợi phản hồi  Một vòng lặp nhận giá trị keepgoing= true để chờ kết nối  Mở luồng kết nối cho client đồng thời thêm client vào listclient. Nếu server bị dừng, luồng đăng nhập của client cũng sẽ bị ngắt |
| 2 | **public** **void** stop() | Giá trị keepgoing=false  Tạo socket mới với localhost và port tương ứng |
| 3 | **private** **synchronized** **void** broadcast(ArrayList<Paint> listPaint) | Gửi thông điệp kiểu broadcast đến tất cả máy client đang tham gia vào phiên  Client nào không gửi được listPaint sẽ bị removed khỏi phiên kết nối |
| 4 | **synchronized** **void** remove(**int** id) | Input vào môt giá trị ID  Duyệt listClient đến khi tìm thấy id  Xóa id khỏi listclient |
| 5 | **public** ClientThread(Socket socket) | Mở luồng socket  Nhận vào username  Gửi ra thông báo username+ just connected |
| 6 | **public** **void** run() | listPaint sẽ nhận giá trị là 1 list đối tượng và vẽ lại. |

1. **Thiết kế giao diện.**

* Sử dụng JFrame (Window contrainer chứa các phần tử như Menubar, toolbar, textfield, và cả JPanel) để thiết kế giao diện.
* Sử dụng WindowBuilder để thiết kế.

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Màn hình, cửa sổ, dialog** | **Mục đích** | **Giải thích** |
| 1 |  | Giao diện chính của manager. | MenuBar: gồm các thao tác như NewFile, OpenFile, SaveFile,…  Toolbar: gồm những button- công cụ để vẽ.  Textfield: Portnumber để nhập số port.  Button Start/Stop:bắt đầu/kết thúc phiên hoạt động.  Dùng class PaintApp khởi tạo môi trường để vẽ.  Button ChooseColor: chọn màu vẽ.  ListUser: chứa tất cả các user đăng nhập vào hệ thống, đồng thời manager có thể quản lý việc hiện diện hoặc loại bỏ của 1 hoặc nhiều user bất kỳ. |
| 2 |  | Giao diện chính của user. | MenuBar: gồm các thao tác như NewFile, OpenFile, SaveFile,…  Toolbar: gồm những button- công cụ để vẽ.  Textfield:  +Server Address: nhập địa chỉ của server.  +Portnumber: nhập số port.  +Username: tên user.  Button:  +Login: đăng nhập vào hệ thống.  +Logout: thoát khỏi hệ thống.  Dùng class PaintApp khởi tạo môi trường để vẽ.  Button  ChooseColor: chọn màu vẽ.  Textfield Messenger: nhập nội dung tin nhắn |

1. **CÀI ĐẶT VÀ KIỂM THỬ.**
2. **Cài đặt.**

* Ngôn ngữ : Java.
* IDE: Esclipes.
* Plugs: WindowBuilder, Swing.
* Mảng chạm đến: JComponent, Socket.io, Serialization Object.

1. **Kiểm thử.**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Tình huống** | **Mục đích** | **Giải thích tình huống hoặc hướng xử lý tình huống** |
| 1 | Thử sử dụng để vẽ, các thao tác vẽ trên form(của cả hai form) | Kiểm tra xem thao tác vẽ có thành công trên form hay không, có vấn đề gì về lỗi vẽ hay không | Vì ứng dụng phải đáp ứng hai yêu cầu chính là cho phép người dùng vẽ và gửi nét vẽ đi nên vẽ được là điều tiên quyết. |
| 2 | Thử mở socket trên Server | Kiểm tra xem có mở socket trên Server được không và có xem xem port có được sử dụng trước đó chưa. | Vì trên một mạng thì các port có thể đã được sử dụng trước đó rồi nên cần check trước ki mở port. |
| 3 | Thử kết nối từ Client đến ServerSocket khi Server chưa mở socket. | Kiểm tra xem có thông báo lỗi khi Server chưa mở socket. | Khi Client truy xuất vào socket của client thì cần phải biết chính xác port cũng như địa chỉ host của Server. |
| 4 | Thử kết nối từ Client đến Server khi đã nhập chính xác địa chỉ IP host của server và port. | Kiểm tra việc quản lý của Server có hiệu quả không bằng việc xác nhận cho Client đăng nhập thành công vào hay ngắt trực tiếp luồng không cho Client truy nhập vào. | Khi “tham gia” vào Server thì Client sẽ được quản trị viên quản trị(Xác nhận đăng nhập, bị từ chối hoặc Ngắt không cho quyền Client đăng nhập nữa). |
| 5 | Thử tắt trực tiếp form Server hoặc Stop Server khi có Client đang kết nối. | Kiểm tra xem Server sẽ “kick out” mọi client ra khỏi server hay không | Khi Server ngưng hoạt động thì mọi Client đang truy xuất server sẽ trở về trạng thái vẽ offline. |
| 6 | Thử tắt trực tiếp form Client khi trực tiếp kết nối với Server. | Kiểm tra xem khi tắt đột ngột Client thì có thông báo logout trên server hay không | Trong quá trình vẽ server cũng sẽ có nhiệm vụ broadcast danh sách nét vẽ cho các client khác nên nếu client thoát ra thì cần phải tắt luồng của Client đó đi. |
| 7 | Thử nhắn tin từ Client khi đã có kết nối đến server. | Kiểm tra ứng dụng chat có chạy không, khi nhắn thì server chỉ xem được nội dung và client mới có quyền nhắn. | Thử xem các serializable object(gồm có ArrayList<Paint> và ChatMessage) có hoạt động chuyển mạch tốt không. |
| 8 | Thử kết nối nhiều Client đến Server khi Server đã mở socket. | Kiểm tra broadcast danh sách nét vẽ hoặc tin nhắn. |  |
| 9 | Thử Open file | Kiểm tra việc mở file ở dạng đuôi .TXT và .PNG. Với TXT thì sẽ lấy lên danh sách nét vẽ, với PNG là fill ảnh. | Fill lên khung vẽ. |
| 10 | Thử Save file | Kiểm tra việc lưu file ở dạng đuôi .TXT và .PNG. | Lưu file. |
| 11 | Thử tính năng show nét vẽ nào của user nào. | Rê chuột qua object hình vẽ của user nào thì hiển thị tên user. | Test tính năng kiếm tra nét vẽ, online. |

1. **KẾT LUẬN**
2. **Kết quả đạt được:**

* Xây dựng ứng dụng vẽ từ xa, tạo môi trường đa người sử dụng với giao diện đơn giản, dễ dàng thao tác cho người mới sử dụng.
* Quản lý đăng nhập, đăng xuất trên server.
* Quản lý được nét vẽ của từng user.
* Nhắn tin giữa các client với nhau thông qua server.

1. **Hạn chế:**

* Quá trình gửi nhận nét vẽ chưa mượt do phải gửi 1 danh sách lớn.(ArrayList<Paint>)
* Khi lưu file và mở file dưới dạng .txt thì chưa đóng gói được phần nét vẽ của username nào.
* Server chưa broadcast chính nó cho các users đang trong phiên.
* Chưa tương tác được với host ngoài mạng LAN.

1. **Hướng phát triền:**

* Tối ưu hóa quá trình gửi nhận bằng class(InfoPaint) chỉ gửi tọa độ điểm không gửi một danh sách.
* Thêm chức năng trao quyền manager cho user.
* Mở rộng dùng internet thay vì chỉ sử dụng mạng LAN đơn thuần.
* Thay vì sử dụng form, ta chuyển sang một ứng dụng web đơn giản.

1. **TÀI LIỆU THAM KHẢO**
2. **Tài liệu tham khảo phần Paint:**

**[1]** [**Java Swing Tutorial**](https://www.javatpoint.com/java-swing)

**[2]** [**Eclipse WindowBuilder Tutorial**](https://examples.javacodegeeks.com/desktop-java/ide/eclipse/eclipse-windowbuilder-tutorial/)

**[3]** [**JComponent (Java Platform SE 7 ) - Oracle Help Center**](https://docs.oracle.com/javase/7/docs/api/javax/swing/JComponent.html)

**[4]** [**Painting JComponent Objects**](http://www.java2s.com/Tutorial/Java/0240__Swing/PaintingJComponentObjects.htm)

**[5] [PDF]** [**Draw, Paint, Repaint**](https://web.stanford.edu/class/archive/cs/cs108/cs108.1092/handouts/27PaintRepaint.pdf)

**[6]** [**Java Examples - BufferedReader and BufferedWriter**](https://owlcation.com/stem/Java-BufferedReader-and-BufferedWriter-Example)

**[7]** [**Save Java component as an image - Technojeeves**](http://technojeeves.com/index.php/73-save-java-component-as-an-image)

**[8]** [**Github - Swing-Paint**](https://github.com/RMuskovets/Swing-Paint.git)

**[9]** [**Github - WhiteBoard**](https://github.com/wybneu/WhiteBoard.git)

**[10]** [**Github - paint**](https://github.com/isac322/paint.git)

1. **Tài liệu tham khảo phần Socket:**

**[1]** [**Java Code Examples io.socket.SocketIO - Program Creek**](https://www.programcreek.com/java-api-examples/?api=io.socket.SocketIO)

**[2]** [**Hướng dẫn lập trình Java Socket - O7planning**](https://o7planning.org/vi/10393/huong-dan-lap-trinh-java-socket)

**[3]** [**io.socket.SocketIO java code examples | Codota**](https://www.codota.com/code/java/classes/io.socket.SocketIO)

**[4]** [**Java - Sockets and Serialization - CodeProject**](https://www.codeproject.com/Tips/991180/Java-Sockets-and-Serialization)

**[5]** [**Youtube - JAVA Sockets**](https://www.youtube.com/watch?v=l4_JIIrMhIQ&list=PLSM8fkP9ppPrRfjQhSakSljzp02Jhg9B0)

**[6]** [**Java Chat Application or Java Messenger - Gadgets Trends**](http://exclusivejava.blogspot.com/2016/06/java-chat-application-or-java-messenger.html)