**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN**

**ĐỒ ÁN 1**

**CHƯƠNG TRÌNH REMOTE DRAW TẠO MÔI TRƯỜNG CHO PHÉP NHIỀU USER CÙNG THỰC HIỆN VẼ TỪ XA**

**GVHD : Ths. Nguyễn Đăng Quang**

**SVTH 1 : Luyện Ngọc Thanh**

**MSSV: 17110221**

**SVTH 2 : Nguyễn Thành Như**

**MSSV:17110202**

**Tp. Hồ Chí Minh, tháng 11** **năm 2019**

**MỤC LỤC**

[LỜI NÓI ĐẦU 1](#_Toc531329805)

[I. PHẦN MỞ ĐẦU 2](#_Toc531329806)

[1. Lí do chọn đề tài 2](#_Toc531329807)

[2. Mục tiêu cho đề tài 2](#_Toc531329808)

[3. Phương pháp nghiên cứu 2](#_Toc531329809)

[4. Giới thiệu đề tài 2](#_Toc531329810)

[II. PHẦN NỘI DUNG 3](#_Toc531329811)

[1. Đặc tả ứng dụng 3](#_Toc531329812)

[1.1. Chức năng ứng dụng 3](#_Toc531329813)

[1.2. Use case diagram của phần mềm 4](#_Toc531329814)

[2. Phân công công việc 4](#_Toc531329816)

[3. Thiết kế 5](#_Toc531329812)

[3.1. Thiết kế lớp 5](#_Toc531329813)

[3.2. Thiết kế giao diện 13](#_Toc531329814)

[4. Cài đặt và kiểm thử 15](#_Toc531329812)

[III. KẾT LUẬN 16](#_Toc531329822)

I[V. TÀI LIỆU THAM KHẢO 17](#_Toc531329823)

**MỤC LỤC HÌNH**

[Hình 1: Giao diện dự kiến của phần mềm. 3](#_Toc531329127)

[Hình 2: Use case diagram RemotePaint. 4](#_Toc531329128)

[Hình 3: Hình ảnh các tools và shapes được chuẩn bị.. 10](#_Toc531329129)

[Hình 4: Giao diện client sau khi thiết kế.. 11](#_Toc531329130)

[Hình 5: Giao diện server sau khi thiết kế.. 11](#_Toc531329131)

# LỜI NÓI ĐẦU

Đồ án 1 là một môn vận dụng những kiến thức cơ bản đã được học ở môn lập trình hướng đối tượng, giúp cho sinh viên ngành Công Nghệ Thông Tin có cái nhìn rõ hơn về lập trình hướng đối tượng bằng việc xây dựng project ứng dụng.

Bằng những kiến thức đã được học nhóm quyết định vận dụng vào xây dựng đề tài: Chương trình remote draw cho phép nhiều user cùng vẽ một bản vẽ từ xa.

# PHẦN MỞ ĐẦU

## Lí do chọn đề tài

Với đề tài “Chương trình remote draw cho phép nhiều user cùng vẽ một bản vẽ từ xa” có thể giúp tác giả vận dụng những kiến thức học được từ môn học lập trình hướng đối tượng, hiểu rõ hơn về object, class và những tính chất của hướng đối tượng. Ngoài ra việc tạo ứng dụng sẽ giúp người dụng có thể sử dụng để phục cho nhu cầu vẽ cơ bản cá nhân hoặc là vẽ nhóm.

## Mục tiêu cho đề tài

Hoàn thành sản phẩm, có thể cho người dụng được các thao tác vẽ cơ bản (vẽ tự do, vẽ các hình khối, xóa nét vẽ, undo và redo thao tác), trải nghiệm phần mềm thành công. Thiết kế được môi trường cho phép nhiều người dùng đồng thời cùng thực hiện được những thao tác trên.

## Phương pháp nghiên cứu

Vận dụng các kiến thức đã học được từ bộ môn “Lập trình hướng đối tượng”, tham khảo các nguồn tài liệu, giáo trình trong nước lẫn ngoài nước. Ngoài ra còn tham khảo thêm các nguồn tài liệu khác trên mạng internet từ đó vận dụng để thực hiện đề tài nghiên cứu một cách thành công nhất.

## Giới thiệu đề tài

Với mục đích phục vụ việc giảng dạy các môn liên quan đến vẽ đồng thời mở rộng hơn là nhiều người đồng thời cùng thao tác vẽ trên một bản vẽ từ xa, đó là lí do bộ phần mềm remote draw đã được ra đời.

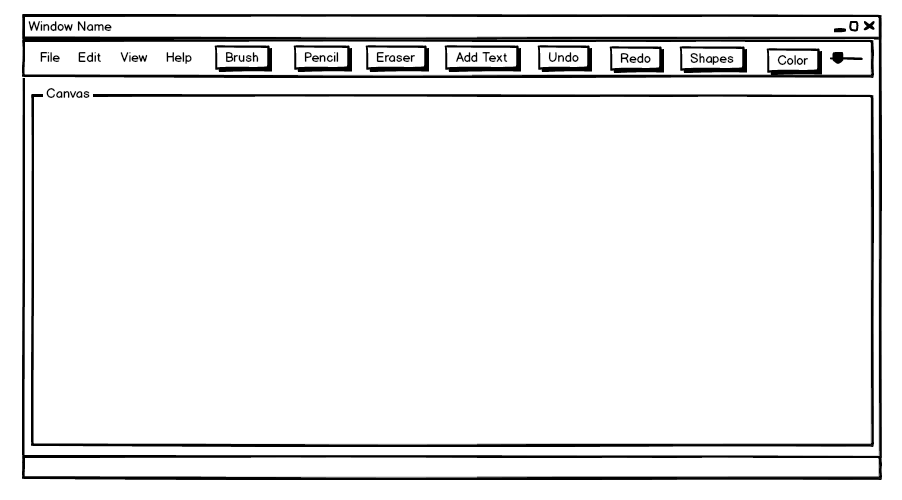
Giới thiệu về bộ phần mềm, bộ phần mềm bao gồm 2 phía là client và server cả 2 đều có đầy đủ chức năng cơ bản của một phần mềm vẽ và khi sử dụng để đồng thời vẽ thì phía server sẽ là nơi nhận các thao tác đến từ phía client.

# PHẦN NỘI DUNG

## Đặc tả ứng dụng

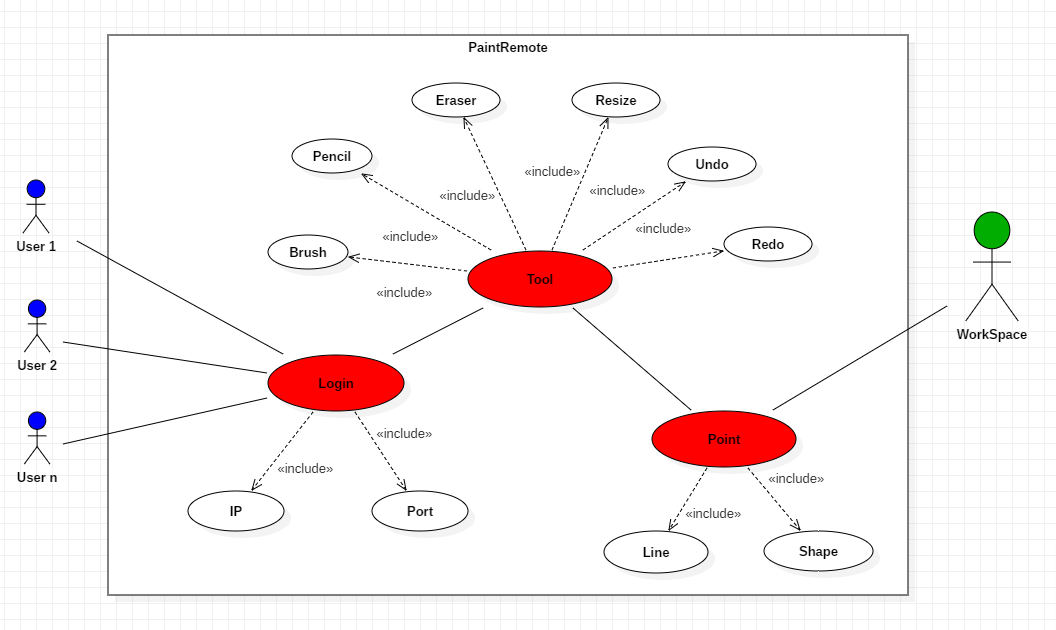
### Chức năng ứng dụng

Là một phần mềm vẽ, phần mềm có những chức năng cơ bản nhất bao gồm: Vẽ tự do, vẽ các đường thẳng các shape cơ bản, xóa nét vẽ và shape, undo redo thao tác mượt mà, thêm text, xuất file ảnh, mở file ảnh sẵn có để chỉnh sửa,…Giao diện phần mềm vô cùng đơn giản thân thiện với người dùng dễ sử dụng không yêu cầu quá cao với người dùng. Đồng thời việc cho phép remote từ xa thông qua mạng LAN, người dùng sẽ có một trải nghiệm mới về việc nhiều người vẽ cùng lúc.



Hình 1: Giao diện dự kiến của phần mềm.

### Use case diagram của phần mềm



Hình : Use case diagram RemotePaint.

## Phân công công việc

|  |  |  |
| --- | --- | --- |
| **Tên công việc** | **Luyện Ngọc Thanh** | **Nguyễn Thành Như** |
| Phân tích đề tài, thiết kế lớp, đối tượng. | **X** | **X** |
| Thiết kế giao diện | **X** |  |
| Code phần mềm vẽ | **X** | **X** |
| Code chức năng | **X** | **X** |
| Kiểm thử phần mềm | **X** | **X** |
| Làm báo cáo | **X** | **X** |
| Thuyết trình sản phẩm | **X** | **X** |
| Mức độ hoàn thành công việc | **100%** | **100%** |

## Thiết kế

### Thiết kế lớp

##### Danh mục các lớp

Sinh viên phụ trách: Luyện Ngọc Thanh

|  |  |  |
| --- | --- | --- |
| **TT** | **Tên Lớp** | **Mục Đích** |
| 1 | Server.java | Thực thi việc mở kết nối cho phép client connect vào, đồng thời khởi tạo các thread đảm nhiệm việc client connect và nhận object từ mỗi client. |
| 2 | Client.java | Thực thi việc connect tới server thông qua ip và port được cung cấp, gửi object tới server và ngắt kết nối tới server. |
| 3 | ConnectionThread và EchoThread nằm trong lớp Server.java | Phân luồng cho mỗi client khác nhau để đồng thời cùng connect tới server |
| 4 | Tools.java | Dùng để xuất dữ liệu trong datagridview ra Excel đã được thiết kế sẵn |
| 5 | PaintFXMLController.java | Là class phục vụ việc điều khiển toàn bộ chương trình. |
| 6 | Interface Draw.java | Thiết lập các method sẵn phục vụ cho việc vẽ các shape |
| 7 | Point.java implements Serializable | Mang tọa độ của vị trí vẽ, nếu client đã connect với server thì sẽ được gửi từ client tới server để server xử lí. |
| 8 | Shape.java implements Serializable | Đại diện cho các shape, dùng để thiết lập các method cho các class con. |
| 9 | Function.cs | Chứa các chức năng bao gồm: tạo mới, mở file, lưu file,undo ,redo. |

Sinh viên phụ trách: Nguyễn Thành Như

|  |  |  |
| --- | --- | --- |
| **TT** | **Tên Lớp** | **Mục Đích** |
| 1 | Client.java | Thực thi việc kết nối từ client tới server thông qua socket TCP/IP, gồm các method phụ trách kết nối và gửi object. |
| 2 | Brush .java | Thiết lập hình ảnh cursor thành cọ vẽ, thiết lập trạng thái vẽ của graphicscontext cho phép người dùng vẽ nét vẽ dạng cọ. |
| 3 | Pencil.java | Thiết lập hình ảnh cursor thành bút chì, thiết lập trạng thái vẽ của graphicscontext cho phép người dùng vẽ nét vẽ dạng bút chì. |
| 3 | Eraser.java | Thiết lập hình ảnh cursor thành tẩy, thiết lập trạng thái vẽ của graphicscontext cho phép người dùng xóa đi nét vẽ |
| 4 | Circles.java extends Shape implements Draw | Vẽ vòng tròn trên canvas. |
| 5 | Rectangles.java extends Shape implements Draw | Vẽ hình chữ nhật trên canvas. |
| 6 | Lines.java extends Shape implements Draw | Vẽ đường thẳng trên canvas. |
| 6 | Ellipses.java extends Shape implements Draw | Vẽ hình ellipse trên canvas. |

Giải thích thêm về các class ở trên:

Client.java

* Mục đích:
* Tạo kết nối từ client đến server thông qua socket TCP/IP.
* Ngắt kết nối tới server.
* Gửi object tới server.

2.Server.java

* Mục đích:
* Khởi tạo một socketserver cho phép các client kết nối vào.
* Mở cho mỗi client một luồng xử lí.
* Xử lí các object được nhận từ client.

#### *Phương thức trong mỗi lớp*

Sinh viên phụ trách: Luyện Ngọc Thanh

* + Các nguồn tham khảo:
  + <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/canvas/Canvas.html>
  + <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/canvas/GraphicsContext.htm>
  + <https://stackoverflow.com/questions/24314668/open-edit-and-save-image-canvas-on-file-pgm-in-javafx>
  + <https://www.geeksforgeeks.org/multi-threaded-chat-application-set-1/>

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên file số thứ tự dòng chứa khai báo |
| 1 | **public Server(int port)**  Input: int port.  Ouput: None. | Là constructor khởi tạo đối tượng Server. | Server.java dòng 33. |
| 2 | **public void RunServer(Canvas canvas)**  Input: Canvas canvas.  Ouput: None. | Chạy thread mở socket để các client connect. | Server.java dòng 37. |
| 3 | **public ConnectionThread(Canvas canvas)**  Input: Canvas canvas. | Là constructor khởi tạo đối tượng ConnectionThread. | Server.java dòng 57 |
| 4 | **public void run()**  Input: None  Ouput:None | Là method của Thread.  Khởi tạo socketserver, chấp nhận yêu cầu kết nối của client, mở thread cho mỗi client. | Server.java dòng 64. |
| 5 | **public EchoThread(Socket socket, ObjectOutputStream out, ObjectInputStream inp, Canvas canvas, ArrayList<Point> list)**  Input: Socket socket, ObjectOutPutSteam out,  ObjectInputStream inp,  Canvas canvas  ArrayList<Point> list.  Output: None | Là constructor khởi tạo đối tượng EchoThread. | Server.java dòng 96 |
| 6 | **public void run()**  Input: None  Ouput: None | Là method của Thread.  Nhận các Point hoặc shape từ client rồi xử lí. | Server.java dòng 106. |
| 7 | **public Client(String ip, int port)**  Input: Chuỗi ip và port  Ouput: None | Là constructor khởi tạo đối tượng client. | Client.java dòng 27. |
| 8 | **public boolean Connect() throws IOException** Input: None  Ouput: True of false | Thực hiện việc connect client tới server, warning nếu server không sẵn có. | Client.java dòng 32. |
| 9 | **public void Disconnect() throws IOException**  Input: none  Ouput:Danh sách liên kết chứa các phần tử của chuỗi trên | Ngắt kết nối client với server. Đảm bảo việc ngắt kết nối không xảy ra lỗi. | Client.java dòng 45. |
| 10 | **public void SendObject(Point p)**  Input: Point p  Ouput: None | Gửi object Point từ client sang server | Client.java dòng 50. |
| 11 | **public void SendObject2(Shape s)**  Input: Shape s  Ouput: None | Gửi object Shape từ client sang server | Client.java dòng 54. |
| 12 | **public void ToolFunction(Canvas canvas,Slider tsize, List<WritableImage> undo, Index i,Client client,ColorPicker color,String author,boolean isConnected)**  Input: Canvas canvas,Slider tsize, List<WritableImage> undo, Index i,Client client,ColorPicker color,String author,boolean isConnected  Output: None | Thiết lập chức năng cho tools, sử dụng các controls như slider, colorpicker,… | Tools.java dòng 33 |
| 13 | **public void UnselectTool(Canvas canvas)**  Input: String name  Output: none | Ngưng sử dụng tool. | Tools.java dòng 30 |
| 14 | **public Shape(Point start,Point end,String shapename,String color,double size,String author)**  Input: Point start,Point end,String shapename,String color,double size,String author | Constructor cho Shape. | Shape.java dòng |
| 15 | **public void setStartPoint(Point start)**  Input: Point start | Thiết lập tọa độ điểm đầu cho shape. | Shape.java dòng |
| 16 | **public void setEndPoint(Point end)**  Input: Point end | Thiết lập tọa độ điểm cuối cho shape. | Shape.java dòng |
| 17 | **public Point getStartPoint()**  Input: none  Output: point | Lấy tọa độ điểm đầu của shape. | Shape.java dòng |
| 18 | **public Point getEndPoint()**  Input: none  Output: point | Lấy tọa độ điểm cuối của shape. | Shape.java dòng |
| 19 | **public void Open(Canvas canvas)**  Input: Canvas canvas  Output: None | Mở file ảnh đã có sẵn để thực hiện vẽ đè. | Function.java dòng |
| 20 | **public void New(Canvas canvas)**  **Input: Canvas canvas**  **Output: none** | Khởi tạo vùng vẽ mới. | Function.java dòng |
| 21 | **public void SaveAs(Canvas canvas)**  **Input: Canvas canvas**  **Output: none** | Lưu vùng vẽ dưới dạng file ảnh. | Function.java dòng |
| 22 | **public void About()**  **Input: none**  **Output: none** | Giới thiệu về đội ngũ thiết kế sản phẩm. | Function.java dòng |
| 23 | **public void Undo(Canvas canvas, List<WritableImage> undo, Index i)** | Thực hiện chức năng undo nét vẽ. | Function.java dòng |
| 24 | **public void Redo(Canvas canvas, List<WritableImage> undo, Index i)** | Thực hiện chức năng redo nét vẽ. | Function.java dòng |
| 25 | **public void Exit()** | Thoát khỏi phần mềm. | Function.java dòng |

Sinh viên phụ trách: Nguyễn Thành Như

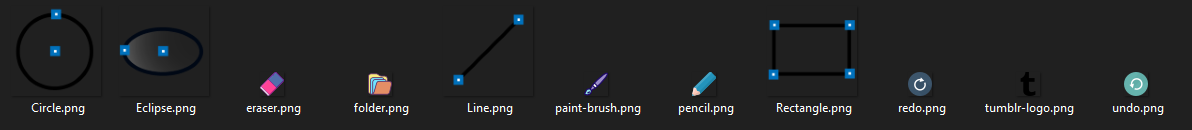
* + Các nguồn tham khảo:
* <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/canvas/Canvas.html>
* <https://docs.oracle.com/javase/8/javafx/api/javafx/scene/canvas/GraphicsContext.htm>

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên file số thứ tự dòng chứa khai báo |
| 1 | **public void autodraw(Canvas canvas,Point start,Point end,String color,double size);** | Thay đổi hình ảnh cursor khi không sử dụng tools, set lại các event cho canvas. | Brush.java dòng 40.  Pencil.java dòng 40.  Eraser.java dòng 40.  AddText.java dòng 50. |
| 2 | **public void stopdraw(Canvas canvas);**Input: Canvas canvas | Thiết lập các event trên canvas cho các tools, lưu hình ảnh tạm phục vụ cho chức năng undo redo và gửi point nếu đã connected với server. | Brush.java dòng 40.  Pencil.java dòng 40.  Eraser.java dòng 40.  AddText.java dòng 50. |
| 3 | **public void SetCursor(Canvas canvas, int tcount)**  Input: Canvas canvas, int tcount  Output: None | Thay đổi hình ảnh cursor khi thực sử dụng tools như brush,pencil,eraser,add text | Brush.java dòng 40.  Pencil.java dòng 40.  Eraser.java dòng 40.  AddText.java dòng 50. |
| 4 | **public void UnselectTool(Canvas canvas)**  Input: Canvas canvas | Thay đổi hình ảnh cursor khi không sử dụng tools, set lại các event cho canvas. | Brush.java dòng 40.  Pencil.java dòng 40.  Eraser.java dòng 40.  AddText.java dòng 50. |
| 5 | **public Tools(String name)**  Input: String name  Output: none | Constructor cho tools. | Tools.java dòng 17 |
| 6 | **public void ToolFunction(Canvas canvas,Slider tsize, List<WritableImage> undo, Index i,Client client,ColorPicker color,String author,boolean isConnected)**  Input: Canvas canvas | Thiết lập các event trên canvas cho các tools, lưu hình ảnh tạm phục vụ cho chức năng undo redo và gửi point nếu đã connected với server. | Brush.java dòng 40.  Pencil.java dòng 40.  Eraser.java dòng 40.  AddText.java dòng 50. |
| 7 | **public Circles(Point start, Point end, String shapename, String color, double ssize,String Author)**  Input: Point start, Point end, String shapename, String color, double ssize,String Author  Output: none | Constructor cho hình tròn. | Circles.java dòng 41 |
| 8 | **public void freedraw(Canvas canvas, Slider tsize, ColorPicker color,List<WritableImage> undo, Index index, Client client,String author,boolean isConnected)**  Input: Point start, Point end, String shapename, String color, double ssize,String Author  Output: none | Để người dùng vẽ các hình. | Draw.java dòng 19  Ellipses.java dòng 77  Lines.java dòng 74  Rectangles.java dòng 80  Circles.java dòng 54 |
| 9 | **public void autodraw(Canvas canvas, Point start, Point end, String color, double size)**  Input: Point start, Point end, String shapename, String color, double ssize,String Author | Dùng để vẽ hình với các tham số được truyền vào. | Draw.java dòng 20  Ellipses.java dòng 50  Lines.java dòng 49  Rectangles.java dòng 48  Circles.java dòng 78 |
| 10 | **public void stopdraw(Canvas canvas)**  input: Canvas canvas  outpu: None | Ngừng việc vẽ hình lại. | Draw.java dòng 21  Ellipses.java dòng 101  Lines.java dòng 91  Rectangles.java dòng 105  Circles.java dòng 97 |
| 11 | **public Ellipses(Point start, Point end, String shapename, String color, double ssize,String Author)**  Input: Point start, Point end, String shapename, String color, double ssize,String Author  Output: None | Constructor cho hình ellipse. | Ellipse.java dòng 40 |
| 12 | **public Rectangles(Point start, Point end, String shapename, String color, double ssize,String author)**  Input: Point start, Point end, String shapename, String color, double ssize,String author  Output: None | Constructor cho hình chữ nhật. | Rectangles.java dòng 41 |
| 13 | **public Lines(Point start, Point end, String name, String color, double ssize,String Author)**  Input: Point start, Point end, String name, String color, double ssize,String Author  Output: None | Constructor cho đường thẳng. | Lines.java dòng 41 |

### Thiết kế giao diện

Việc thiết kế giao diện của ứng dụng là một phần rất quan trọng trong việc xây dựng ứng dụng, là bộ mặt của ứng dụng, nơi người dùng trực tiếp thao tác với ứng dụng. Tạo sự hứng thú, thu hút cho người dùng.

#### Chuẩn bị hình ảnh



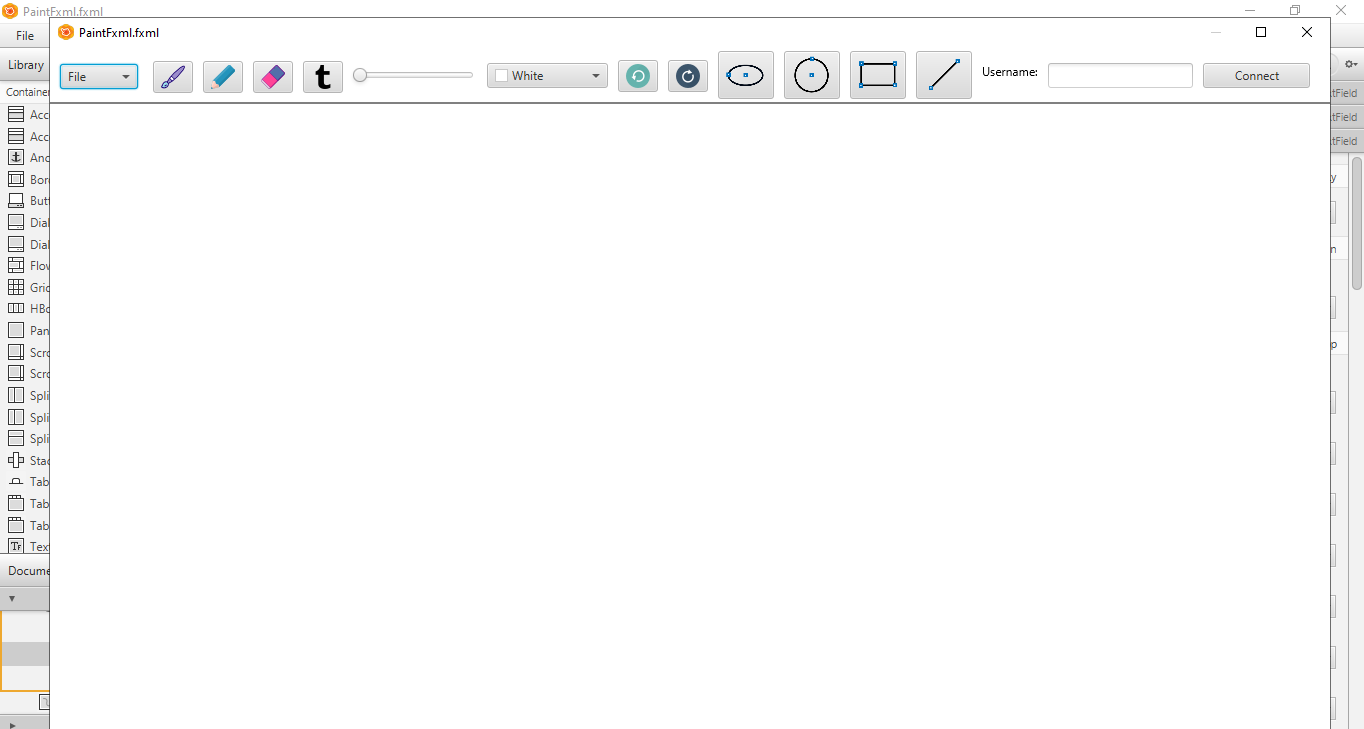
Hình : Hình ảnh các tools và shapes được chuẩn bị.

* Tools: Có bốn tool bao gồm cọ vẽ (brush), chì (pencil), tẩy (eraser), thêm chữ (add text).
* Functions: Có nhiều function nhưng ở phần mềm vẽ thì hoàn tác (undo), tái hoàn tác (redo) là 2 function chính.
* Thay đổi hình ảnh trỏ chuột thành hình ảnh của tools: Đây cũng là một điểm thú vị giúp phần mềm trở nên thân thiện, giảm độ nhàm chán cho người dùng.

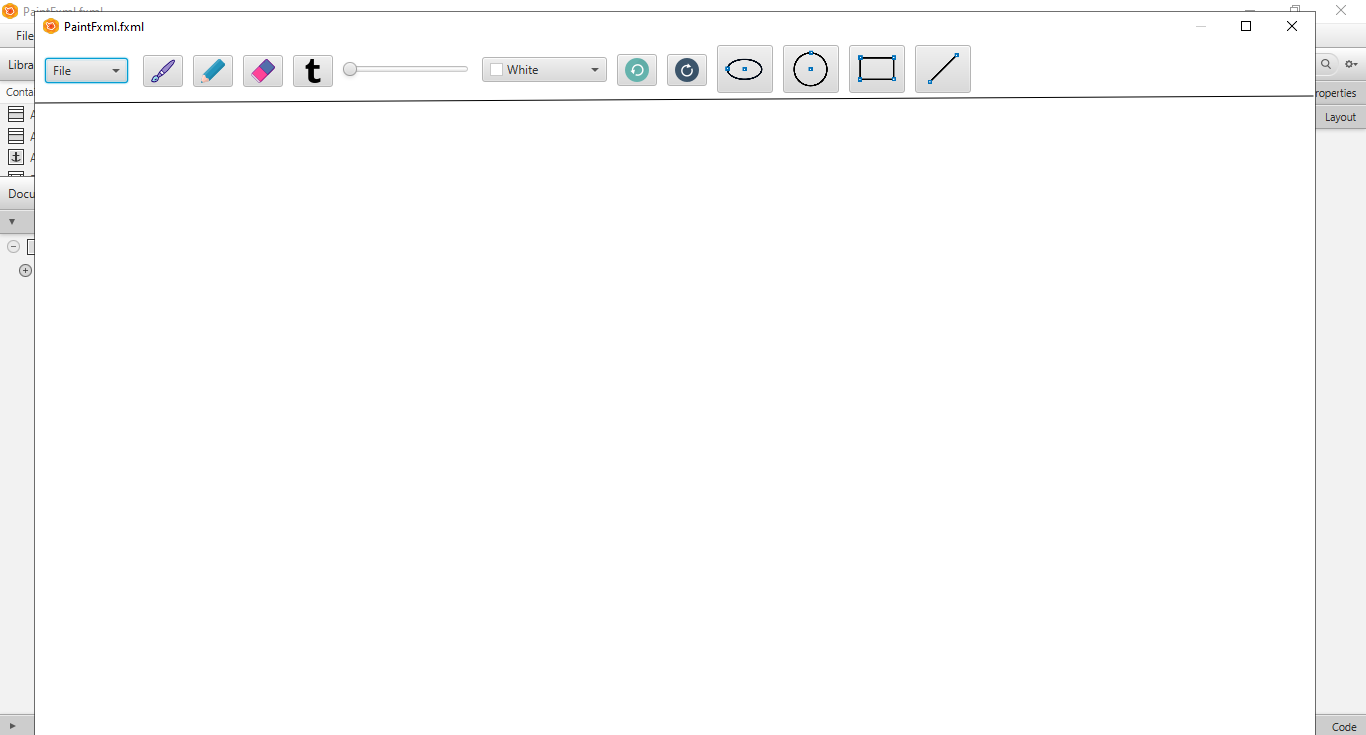
#### Thiết kế giao diện với JavaFX

Kể từ lúc này chúng ta sẽ bắt đầu sử dụng JavaFX với scene builder để thiết kế và lập trình ứng dụng.

* Sau khi tạo project thành công, chúng ta bắt đầu điều chỉnh pane với các thao tác kéo thả chuột và thay đổi giá trị ở phần layout và properties, cố định size cho pane bằng code ở file chính project. Tạo vbox, hbox ( tương tự như panel) để làm thanh công cụ, thêm các button, thay đổi mục *text* và set id cho từng button, thêm slider để chỉnh size cho nét vẽ và hình vẽ, thêm colorpicker để chọn màu.
* Đối với client, thêm textfield để nhập username và button connect để connect tới server.
* Thêm canvas là vùng để vẽ vào project.
* Thêm hình ảnh tools và shapes từ thư mục vào *Source Packages*.
* Tạo các *PictureBox* trong thanh *Toolbox* và set làm children của các button để hiện thị hình ảnh cho tools và shapes.
* Tiến hành tinh chỉnh vị trí cho các elements.



Hình 4: Giao diện client sau khi thiết kế.



Hình : Giao diện server sau khi thiết kế.

## Cài đặt và kiểm thử

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tình huống | Mục đích | Giải thích ngắn gọn quyết định lựa chọn và tình huống đầu vào |
| 1 | Nhiều client cùng connect vào server và đồng thời vẽ. | Kiểm tra xem server có thể nhận được cùng lúc từ nhiều nguồn không. | Vì theo yêu cầu là nhiều user đồng thời thao tác vẽ nên cần phải kiểm tra tình huống này trước tiên. |
| 2 | Client connect tới server khi server chưa khởi động. | Chắc chắn việc connect giữa client và server. | Client có thể vẽ mà không cần connect tới server và chỉ connect khi user muốn. |
| 3 | Client disconnect sau đó connect lại tới server. | Đảm bảo việc người dùng tùy ý connect hay disconnect tới server. | Sau khi connect và vẽ người dùng không muốn vẽ trên server nữa thì có thể ngắt và nếu muốn vẽ tiếp thì connect lại. |

# KẾT LUẬN

* Mức độ hoàn thành mục tiêu: 90%.
* Khó khăn – khắc phục

|  |  |
| --- | --- |
| Khó khăn gặp phải | Cách khắc phục |
| Nhận object từ nhiều client. | Phân luồng riêng cho từng class, mỗi luồng đảm nhận việc nhận object và xử lí. |
| SerialVersionUID của client khác server. | Cài đặt biết static final SerialVersionUID cho client và server. |
| Undo redo khi không kết nối server. | Với mỗi khi nét vẽ hoàn thành, lưu toàn bộ hình ảnh trên canvas vào 1 list sau đó sử dụng một biến cờ toàn cục để thao tác undo redo trên list. |
| Undo redo khi kết nối server. | Chưa tìm ra giải pháp. |

* Ưu điểm của đồ án: Đồ án đã có đủ những tính năng cơ bản của một phần mềm vẽ, tính năng cho phép nhiều người dùng đồng thời thao tác trên canvas đã có.
* Những khuyết điểm của đồ án: so với các sản phẩm tương tự khác bên ngoài thì sản phẩm hiện tại vẫn còn chưa phát triển hoàn thiện nhất về các tính năng (có thể thêm các tính năng mới), giao diện chưa thật sự quá bắt mắt, cầu kì bằng các sản phẩm khác.
* Hướng phát triển đồ án: có thể thêm phát triển đồ án thành một đồ án cho phép remote thông qua mạng internet từ đó việc sử dụng cùng lúc bởi nhiều người dùng ở nhiều khoảng cách địa lí khác nhau sẽ trở nên vô cùng dễ dàng.

# TÀI LIỆU THAM KHẢO

[1] Javafx: <https://docs.oracle.com/javase/8/javafx/api/index.html>.

[2] Java Networking: <https://www.tutorialspoint.com/java/java_networking.htm>.

Video tham khảo cho đề tài: Video hướng dẫn “JavaFX Software Tutorial: Paint (MVC)” của tác giả: Almas Baimagambetov.