**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

[**LỜI NÓI ĐẦU 1.**](#_Toc531689439)

**I. Đặc tả** [**2.**](#_Toc531689440)

[**1. Giới thiệu chung về đề tài 2.**](#_Toc531689442)

[**1.1. Giới thiệu đề tài 2.**](#_Toc531689501)

[**1.2. Mô hình đối tượng 2.**](#_Toc531689501)

[**1.3. Các chức năng 2.**](#_Toc531689501)

[**2. Tìm hiểu về Java 3.**](#_Toc531689442)

[**2.1. Ngôn ngữ Java 3.**](#_Toc531689451)

[**2.2. Môi trường Java Swing 3.**](#_Toc531689451)

[**2.3. Tìm hiểu về SocketIO 3.**](#_Toc531689451)

**II. Phân công công việc** [**3.**](#_Toc531689440)

**III. Thiết kế** [**3.**](#_Toc531689440)

[**1. Thiết kế sơ đồ UseCase 3.**](#_Toc531689501)

[**2. Thiết kế sơ đồ Diagram 5.**](#_Toc531689501)

[**2.1. Sơ đồ tổng thể ứng dụng**](#_Toc531689451) **6.**

[**2.2. Sơ đồ thiết kế Shape**](#_Toc531689451) **6.**

[**2.3. Sơ đồ thiết kế saveFile**](#_Toc531689451) **6.**

[**2.4. Sơ đồ thiết kế WhiteboardClient**](#_Toc531689451) **7.**

[**2.5. Sơ đồ thiết kế WhiteboardServer**](#_Toc531689451) **8.**

[**2.6. Sơ đồ thiết kế SocketClient**](#_Toc531689451) **8.**

[**2.7. Sơ đồ thiết kế SocketServer**](#_Toc531689451) **9.**

[**2. Thiết kế lớp**](#_Toc531689501) **9.**

[**2.1. Bảng danh mục các lớp sử dụng trong chương trình**](#_Toc531689451) **9.**

[**2.2. Bảng mô tả phương thức trong một lớp 14.**](#_Toc531689451)

[**3. Thiết kế giao diện 16.**](#_Toc531689501)

**V. Kết luận** [**19.**](#_Toc531689440)

**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.

**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 save:* Lưu hình ảnh dưới dạng file .png hoặc .txt
3. *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.
4. Chức năng chấp nhận phiên hoạt động của user khi đăng nhập vào hệ thống.
5. Chức năng xóa người dùng phiên hoạt động của user.
6. Chức năng chọn màu của bút vẽ.
7. Chức năng mở một file đã lưu về.
8. **Tìm hiểu ngôn ngữ Java và môi trường Java Swing.**
   * 1. **Ngôn ngữ Java.**
     2. **Môi trường Java Swing.**
     3. **Tìm hiểu SocketIO**
9. **PHÂN CÔNG CÔNG VIỆC.**

|  |  |  |  |
| --- | --- | --- | --- |
| **TT** | **Tên sinh viên** | **Mô tả khái quát công việc sinh viên thực hiện trong dự án** | **Phần trăm đóng góp** |
| 1 | Trần Minh Hiếu | Package Frame, Socket, Paint  Xây dựng sơ đồ use case  Thiết kế sơ đồ Diagram | 100% |
| 2 | Nguyễn Ngọc Tú Duyên | Package Object  Viết báo cáo  Xây dựng sơ đồ use case | 100% |
| 3 | Võ Thị Vân Anh | Package Shape  Viết báo cáo  Xây dựng sơ đồ use case | 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.



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



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

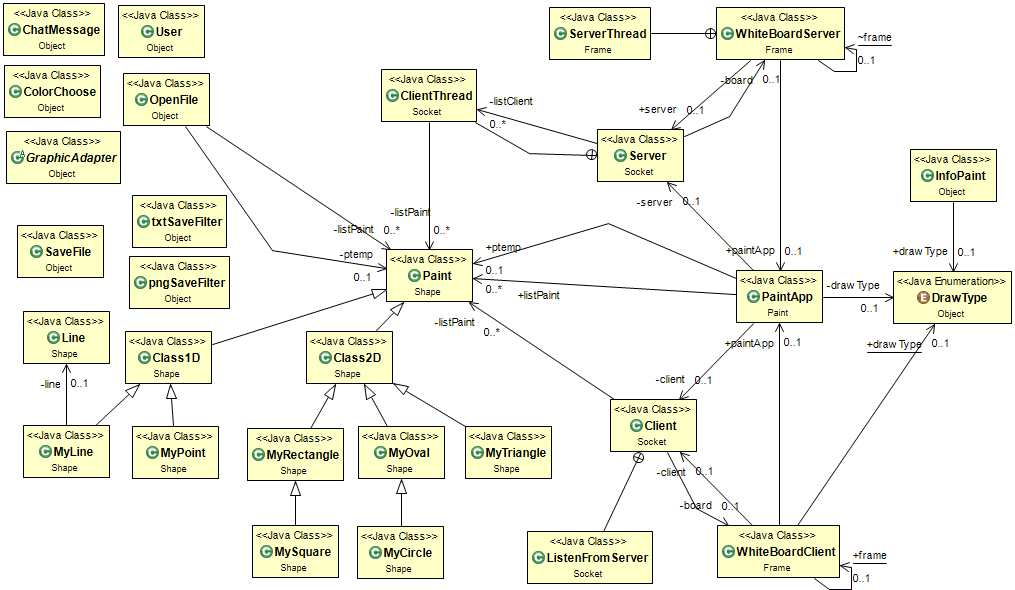


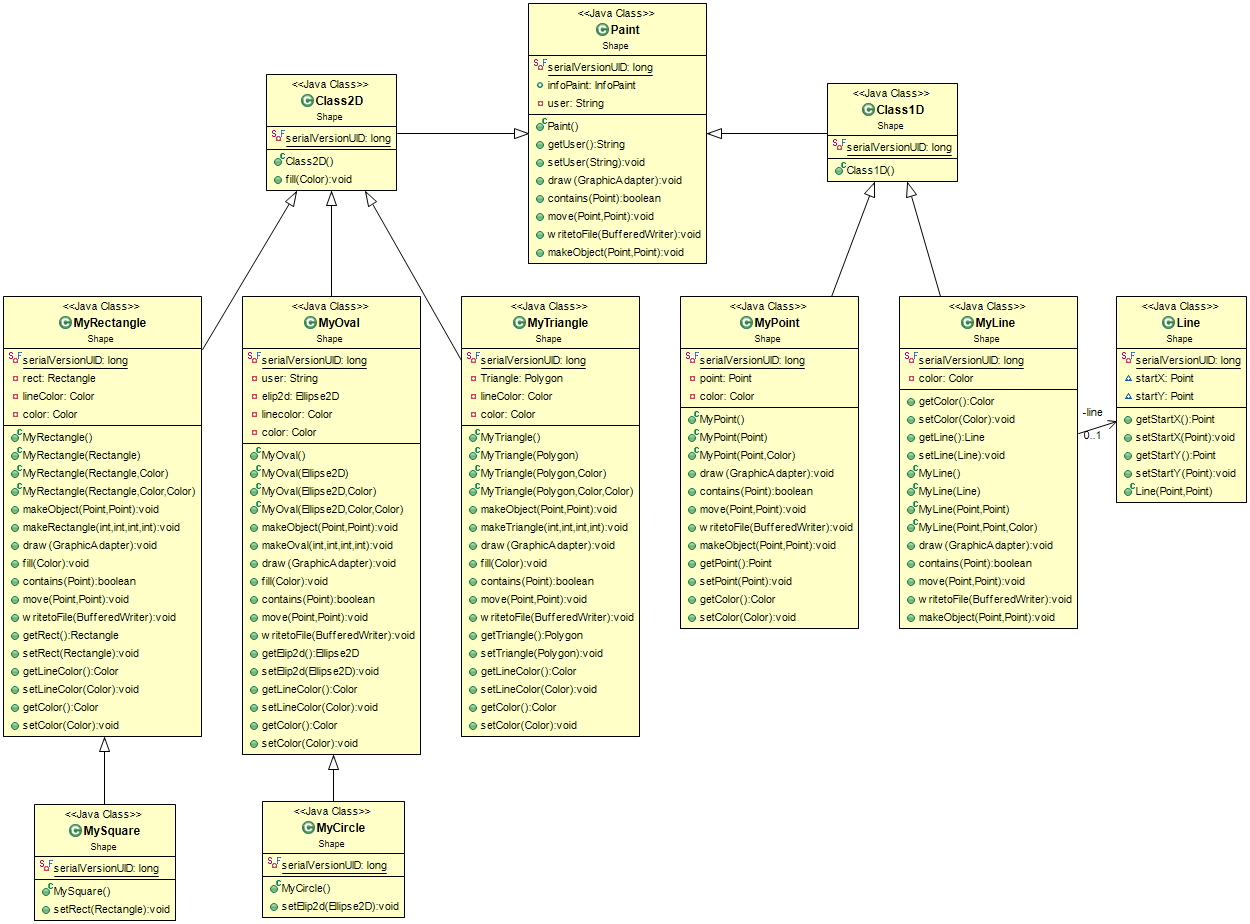
Không gian vẽ bao gồm quản lý nét vẽ của mỗi người dùng, lưu phiên làm việc, quản lý phiên làm việc.



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 Diagram.**
   1. **Sơ đồ tổng thể ứng dụng.**

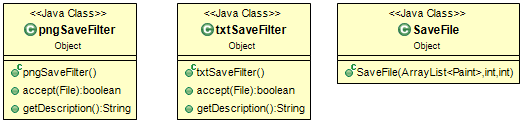
****

* 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.

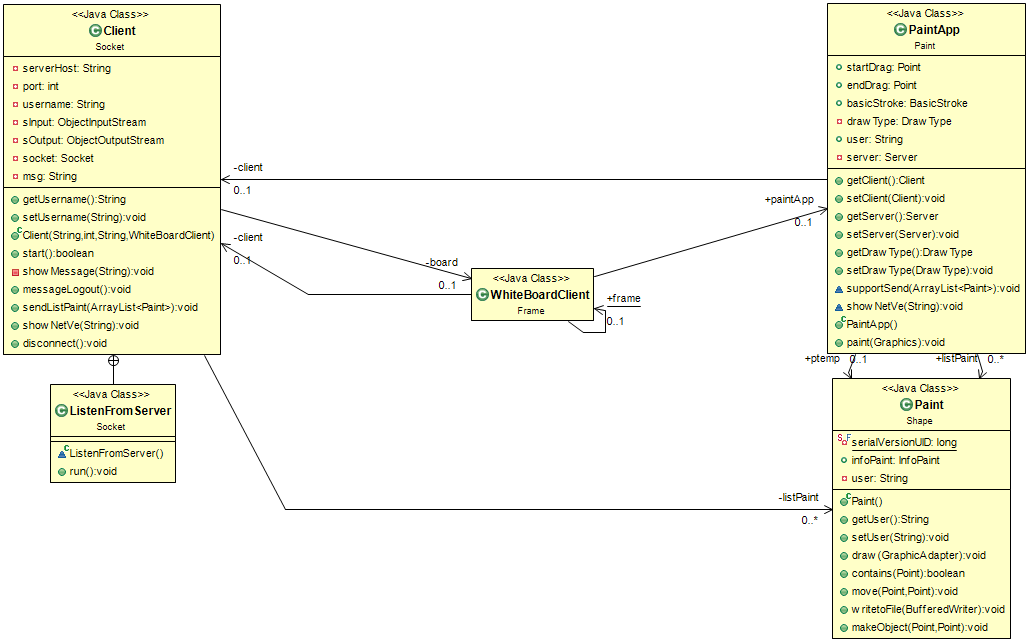
Từ Class1D ta tạo ra 2 class kế thừa đến là MyPoint và MyLine, Class Line khi dùng sẽ gọi lại class MyLine.

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.

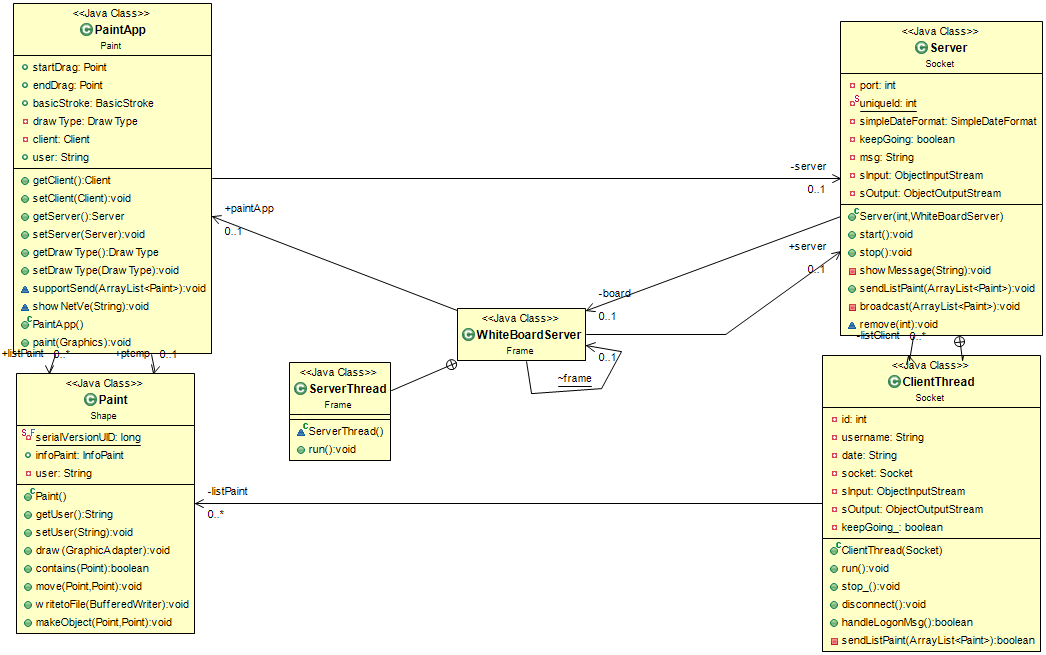
* 1. **Sơ đồ thiết kế WhiteboardClient.**



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.**

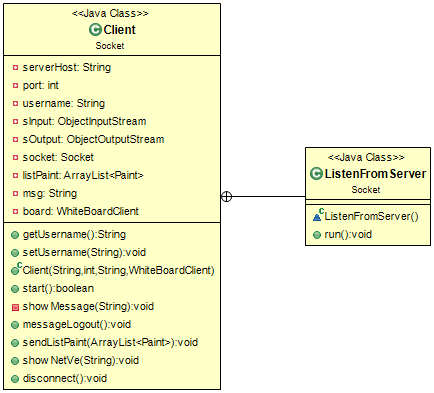


Đầu tiên, manager có thể mở và dùng ứng dụng offline hoặc online (thông qua socket), WhiteboardServer sẽ trả về 1 giao diện vẽ với các đối tượng để vẽ.

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.

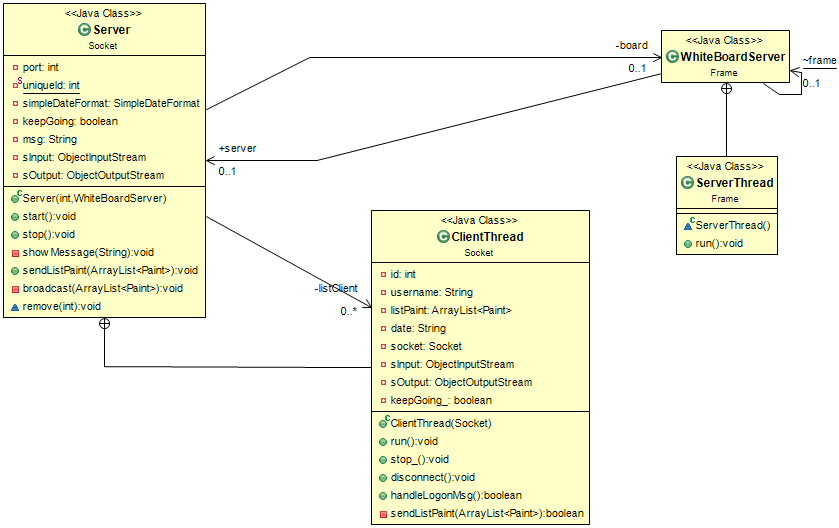
Vì Server có quyền quản lý lên user nên Server có trách nhiệm mở luồng (ServerThread) và nhận luồng (ClientThread).

* 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.**
   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. **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 điểm và vẽ đường thẳng |
| 3 | Class MyPoint | Được kế thừa từ Class1D  Là lớp vẽ điểm |
| 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. **Paint.**

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.

|  |  |  |  |
| --- | --- | --- | --- |
| **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. **KẾT LUẬN**

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 được nét vẽ của từng user.

Hạn chế:

* Quá trình gửi nhận nét vẽ chưa mượt
* Server chưa broadcast chính nó cho các users đang trong phiên

Hướng phát triền:

* Tối ưu hóa quá trình gửi nhận
* Thêm chức năng trao quyền manager cho user