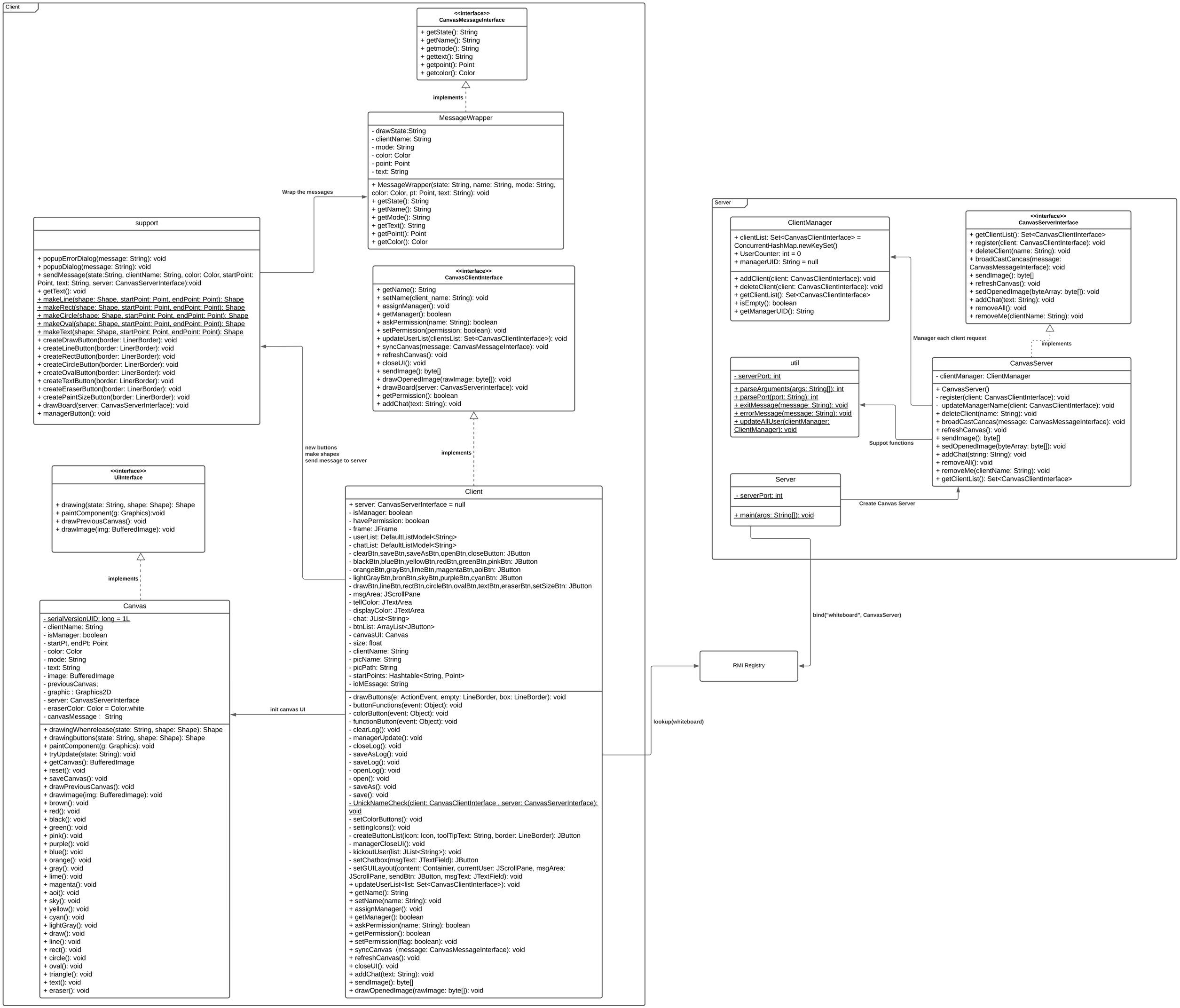
Client
<<interface>>
CanvasMessageInterface
+ getState(): String
+ getName(): String
+ getmode(): String
+ gettext(): String
+ getpoint(): Point
+ getcolor(): Color
implements
MessageWrapper
- drawState:String
- clientName: String
- mode: String
- color: Color
- point: Point
- text: String
+ MessageWrapper(state: String, name: String, mode: String, color: Color, pt: Point, text: String): void
+ getState(): String
+ getName(): String
+ getMode(): String
+ getText(): String
+ getPoint(): Point
+ getColor(): Color
Wrap the messages
support
+ popupErrorDialog(message: String): void
+ popupDialog(message: String): void
+ sendMessage(state:String, clientName: String, color: Color, startPoint: Point, text: String, server: CanvasServerInterface):void
+ getText(): void
+ makeLine(shape: Shape, startPoint: Point, endPoint: Point): Shape
+ makeRect(shape: Shape, startPoint: Point, endPoint: Point): Shape
+ makeCircle(shape: Shape, startPoint: Point, endPoint: Point): Shape
+ makeOval(shape: Shape, startPoint: Point, endPoint: Point): Shape
+ makeText(shape: Shape, startPoint: Point, endPoint: Point): Shape
+ createDrawButton(border: LinerBorder): void
+ createLineButton(border: LinerBorder): void
+ createRectButton(border: LinerBorder): void
+ createCircleButton(border: LinerBorder): void
+ createOvalButton(border: LinerBorder): void
+ createTextButton(border: LinerBorder): void
+ createEraserButton(border: LinerBorder): void
+ createPaintSizeButton(border: LinerBorder): void
+ drawBoard(server: CanvasServerInterface): void
+ managerButton(): void
<<interface>>
CanvasClientInterface
+ getName(): String
+ setName(client_name: String): void
+ assignManager(): void
+ getManager(): boolean
+ askPermission(name: String): boolean
+ setPermission(permission: boolean): void
+ updateUserList(clientsList: Set<CanvasClientInterface>): void
+ syncCanvas(message: CanvasMessageInterface): void
+ refreshCanvas(): void
+ closeUI(): void
+ sendImage(): byte[]
+ drawOpenedImage(rawImage: byte[]): void
+ drawBoard(server: CanvasServerInterface): void
+ getPermission(): boolean
+ addChat(text: String): void
new buttons
make shapes
send message to server
implements
<<interface>>
UiInterface
+ drawing(state: String, shape: Shape): Shape
+ paintComponent(g: Graphics):void
+ drawPreviousCanvas(): void
+ drawImage(img: BufferedImage): void
implements
Client
+ server: CanvasServerInterface = null
- isManager: boolean
- havePermission: boolean
- frame: JFrame
- userList: DefaultListModel<String>
- chatList: DefaultListModel<String>
- clearBtn,saveBtn,saveAsBtn,openBtn,closeButton: JButton
- blackBtn,blueBtn,yellowBtn,redBtn,greenBtn,pinkBtn: JButton
- orangeBtn,grayBtn,limeBtn,magentaBtn,aoiBtn: JButton
- lightGrayBtn,bronBtn,skyBtn,purpleBtn,cyanBtn: JButton
- drawBtn,lineBtn,rectBtn,circleBtn,ovalBtn,textBtn,eraserBtn,setSizeBtn: JButton
- msgArea: JScrollPane
- tellColor: JTextArea
- displayColor: JTextArea
- chat: JList<String>
- btnList: ArrayList<JButton>
- canvasUI: Canvas
- size: float
- clientName: String
- picName: String
- picPath: String
- startPoints: Hashtable<String, Point>
- ioMEssage: String
- drawButtons(e: ActionEvent, empty: LineBorder, box: LineBorder): void
- buttonFunctions(event: Object): void
- colorButton(event: Object): void
- functionButton(event: Object): void
- clearLog(): void
- managerUpdate(): void
- closeLog(): void
- saveAsLog(): void
- saveLog(): void
- openLog(): void
- open(): void
- saveAs(): void
- save(): void
- UnickNameCheck(client: CanvasClientInterface , server: CanvasServerInterface): void
- setColorButtons(): void
- settingIcons(): void
- createButtonList(icon: Icon, toolTipText: String, border: LineBorder): JButton
- managerCloseUI(): void
- kickoutUser(list: JList<String>): void
- setChatbox(msgText: JTextField): JButton
- setGUILayout(content: Containier, currentUser: JScrollPane, msgArea: JScrollPane, sendBtn: JButton, msgText: JTextField): void
+ updateUserList<list: Set<CanvasClientInterface>): void
+ getName(): String
+ setName(name: String): void
+ assignManager(): void
+ getManager(): boolean
+ askPermission(name: String): boolean
+ getPermission(): boolean
+ setPermission(flag: boolean): void
+ syncCanvas (message: CanvasMessageInterface): void
+ refreshCanvas(): void
+ closeUI(): void
+ addChat(text: String): void
+ sendImage(): byte[]
+ drawOpenedImage(rawImage: byte[]): void
init canvas UI
Canvas
- serialVersionUID: long = 1L
- clientName: String
- isManager: boolean
- startPt, endPt: Point
- color: Color
- mode: String
- text: String
- image: BufferedImage
- previousCanvas;
- graphic : Graphics2D
- server: CanvasServerInterface
- eraserColor: Color = Color.white
- canvasMessage： String
+ drawingWhenrelease(state: String, shape: Shape): Shape
+ drawingbuttons(state: String, shape: Shape): Shape
+ paintComponent(g: Graphics): void
+ tryUpdate(state: String): void
+ getCanvas(): BufferedImage
+ reset(): void
+ saveCanvas(): void
+ drawPreviousCanvas(): void
+ drawImage(img: BufferedImage): void
+ brown(): void
+ red(): void
+ black(): void
+ green(): void
+ pink(): void
+ purple(): void
+ blue(): void
+ orange(): void
+ gray(): void
+ lime(): void
+ magenta(): void
+ aoi(): void
+ sky(): void
+ yellow(): void
+ cyan(): void
+ lightGray(): void
+ draw(): void
+ line(): void
+ rect(): void
+ circle(): void
+ oval(): void
+ triangle(): void
+ text(): void
+ eraser(): void
lookup(whiteboard)
Server
ClientManager
+ clientList: Set<CanvasClientInterface> = ConcurrentHashMap.newKeySet()
+ UserCounter: int = 0
+ managerUID: String = null
+ addClient(client: CanvasClientInterface): void
+ deleteClient(client: CanvasClientInterface): void
+ getClientList(): Set<CanvasClientInterface>
+ isEmpty(): boolean
+ getManagerUID(): String
<<interface>>
CanvasServerInterface
+ getClientList(): Set<CanvasClientInterface>
+ register(client: CanvasClientInterface): void
+ deleteClient(name: String): void
+ broadCastCancas(message: CanvasMessageInterface): void
+ sendImage(): byte[]
+ refreshCanvas(): void
+ sedOpenedImage(byteArray: byte[]): void
+ addChat(text: String): void
+ removeAll(): void
+ removeMe(clientName: String): void
Manager each client request
implements
util
- serverPort: int
+ parseArguments(args: String[]): int
+ parsePort(port: String): int
+ exitMessage(message: String): void
+ errorMessage(message: String): void
+ updateAllUser(clientManager: ClientManager): void
CanvasServer
- clientManager: ClientManager
+ CanvasServer()
- register(client: CanvasClientInterface): void
- updateManagerName(client: CanvasClientInterface): void
+ deleteClient(name: String): void
+ broadCastCancas(message: CanvasMessageInterface): void
+ refreshCanvas(): void
+ sendImage(): byte[]
+ sedOpenedImage(byteArray: byte[]): void
+ addChat(string: String): void
+ removeAll(): void
+ removeMe(clientName: String): void
+ getClientList(): Set<CanvasClientInterface>
Suppot functions
Server
- serverPort: int
+ main(args: String[]): void
Create Canvas Server
bind("whiteboard", CanvasServer)
RMI Registry