Final Project: Chatter UML Diagrams

ServerMain	
-PORT: int	The port the server will listen on
+main(String[] args): void	The entry point to start the server

Server	
-port: int	The port the server will listen on
-serverWorkers: Set	Holds all the current ServerWorkers running
-userNames: Set	Holds all the current users logged in
-db: JDBC	Creates a private JDBC class
+Server(port: int)	Constructs a ServerSocket with the given port number
+run(): void	Connects to the db and handles client connections
+getServerWorkers(): Set	Returns a set of ServerWorkers
+getUserNames(): Set	Returns a set of UserNames
+isValidUser(userName: String): boolean	Returns true of the supplied userName is in the db and is not logged in
+isLoggedIn(userName: String): boolean	Returns true if the userName is in the set of userNames
+addUserName(userName: String): void	Adds the userName to the set of userNames
+removeUser(serverWorker: ServerWorker): void	Removes the serverWorker and userName from their respective sets

JDBC	
-connection: Connection	The connection to the database
+runDB(): void	Connects to the db, creates the db, users table, and adds the users if not already created
+closeDB(): void	Closes the connection to the database
+isUser(name: String): boolean	Returns true if the supplied name is in the db

ServerWorker	
-userSocket: Socket	The clients Socket connection
-server: Server	An instance of the Server class
-userName: String	The clients userName
-out: OutputStream	The output stream of the socket
-reader: BufferedReader	The input reader of the socket
+ServerWorker(socket: Socket, server: Server)	Initializes a new ServerWorker with the supplied socket and server
+run(): void	Initializes I/O streams, and closes socket once user logs out
-handleLogin(): boolean	Gets the clients userName and validates it against Server and db
-handleInput(): void	Handles the input and output streams between the client and server
-getOnlineUsers(): String	Returns a list of online users and outputs to client's terminal
-sendPrivate(message: String[]): void	Sends a private message to the supplied user
-broadcast(message: String): void	Broadcasts a message to all users currently connected
-send(message: String): void	Writes the supplied message to the output stream
-setUserName(userName: String): void	Sets the userName of the client
+getUserName(): String	Returns the clients user name
-removeUser(): void	Removes the client from the server
-isUser(name: String): ServerWorker	Returns the ServerWorker thread of the supplied user name