Phase 2 and 3 Socket Program Design Specification

Group Members: Aimable, Javon, Evan, Femi, Toni

**Network Architecture Design**

**Server Components**

**MainServer - Primary server component responsible for initializing and managing the chat server infrastructure.**

* Main Responsibilities
  + Server socket initialization and port binding
  + Client connection acceptance
  + Thread pool management
  + Resource allocation and deallocation
  + System shutdown coordination
* Core server class that initializes the server socket and manages client connections
* Listens for incoming connections on a specified port
* Maintains a list of connected clients (ClientHandler instances)
* Implements thread pooling for handling multiple clients efficiently

Key Methods for Implementation

* void initialize() // Setup server resources
* void acceptConnections() // Handle incoming connections
* void broadcastMessage() // Send message to all clients
* void removeClient() // Handle client disconnection
* void shutdown() // Graceful server shutdown

**ClientHandler - Manages individual client connections and their communication with the server.**

* Manages individual client connections
* Runs in its own thread to handle client communication
* Maintains client metadata (username, IP address, connection status)
* Responsible for message broadcasting and private messaging

Key Methods for Implementation

* void processMessages() // Handle incoming messages
* void sendMessage() // Send message to client
* void handleDisconnect() // Manage client disconnection
* void validateMessage() // Verify message integrity

**MessageProcessor - Handles message formatting, validation, and routing between clients.**

* Main Responsibilities  
  + Message parsing and formatting
  + Message validation
  + Message routing
  + Error handling
  + Message queuing
* Handles message formatting and processing
* Implements message types (broadcast, private, system notifications)
* Manages message queuing and delivery confirmation

**Client Components**

**ChatClient- Manages the client's connection to the server and handles user interactions.**

* Main Component Responsibilities
  + Server connection management
  + Message sending/receiving
  + User interface updates
  + Error handling
  + Connection state management
* Manages connection to server
* Maintains local state (connection status, message history)
* Implements reconnection logic on connection loss
* Key Methods for Implementation
  + Key Responsibilities
  + Server connection management
  + Message sending/receiving
  + User interface updates
  + Error handling
  + Connection state management

**MessageListener- Dedicated component for handling incoming messages from the server.**

* Main Key Responsibilities
  + Message reception
  + Connection monitoring
  + UI update triggering
  + Error detection
* Runs in separate thread
* Listens for incoming messages from server
* Updates GUI with received messages
* Handles server disconnection events

**Phase 3: Setup and Prototyping**

**Code in Attached Zipfile**

**Detailed Team Member Responsibilities**

**Aimable M. (Team Leader)**

**Primary Focus: Core Server Development & Project Coordination**

1. **Server Core Development**
   * Implement MainServer class including:
     + Server socket initialization and configuration
     + Thread pool management system
     + Client connection acceptance logic
     + Resource allocation/deallocation mechanisms
2. **Project Coordination**
   * Setup and maintain project structure:
   * Create development milestones and deadlines
   * Ensure component integration
   * Monitor progress and adjust assignments
3. **GitHub Repository Management**
   * Setup repository structure
   * Maintain documentation

**Evan**

**Primary Focus: Network Protocol Implementation**

1. **TCP Socket Implementation**
   * Develop MessageProcessor class
   * Implement message routing system
   * Create connection pooling mechanism
2. **Protocol Development**
   * Define message formats
   * Implement message validation
   * Create error handling protocols
3. **API Documentation**
   * Document all network-related methods
   * Create sequence diagrams for message flow
   * Provide integration guidelines

**Javon**

**Primary Focus: Client GUI Development**

1. **Client Interface Implementation**
   * Develop ClientGUI components
   * Implement event handlers
   * Create input validation system
2. **GUI Features**
   * Message display formatting
   * User list management
   * Connection status indicators
   * Error message displays
3. **Client-Side Integration**
   * Connect GUI to network layer
   * Implement message sending/receiving
   * Create client-side error handling

**Femi**

**Primary Focus: Server GUI Development**

1. **Server Interface Implementation**
   * Develop ServerGUI components
   * Create monitoring displays
   * Implement admin controls
2. **Documentation**
   * Create user manual with screenshots
   * Document server configuration
   * Provide setup instructions
   * Create troubleshooting guide
3. **Integration Testing**
   * Test GUI-Server integration
   * Verify admin controls
   * Validate monitoring systems

**Toni**

**Primary Focus: Testing & Analysis**

1. **Testing Framework**
   * Create test scenarios
   * Implement automated tests
   * Create performance tests
2. **Wireshark Analysis**
   * Setup capture configurations
   * Define test scenarios
   * Document packet flows
3. **Error Handling**
   * Implement validation
   * Create error logging system
   * Implement recovery procedures

Phase 2 and 3 Socket Program Design Specification

Group Members: Aimable, Javon, Evan, Femi, Toni