Remote Desktop System

Architecture & Component Flow Diagram

System Overview

The Remote Desktop System is a client-server application that allows remote screen viewing, control, and collaboration. The system provides screen sharing, remote control, chat functionality, and file transfer capabilities, with all activities tracked in a MongoDB database.

Screen Sharing

Capture and transmit screen contents to connected clients with customizable quality and frame rate.

Remote Control

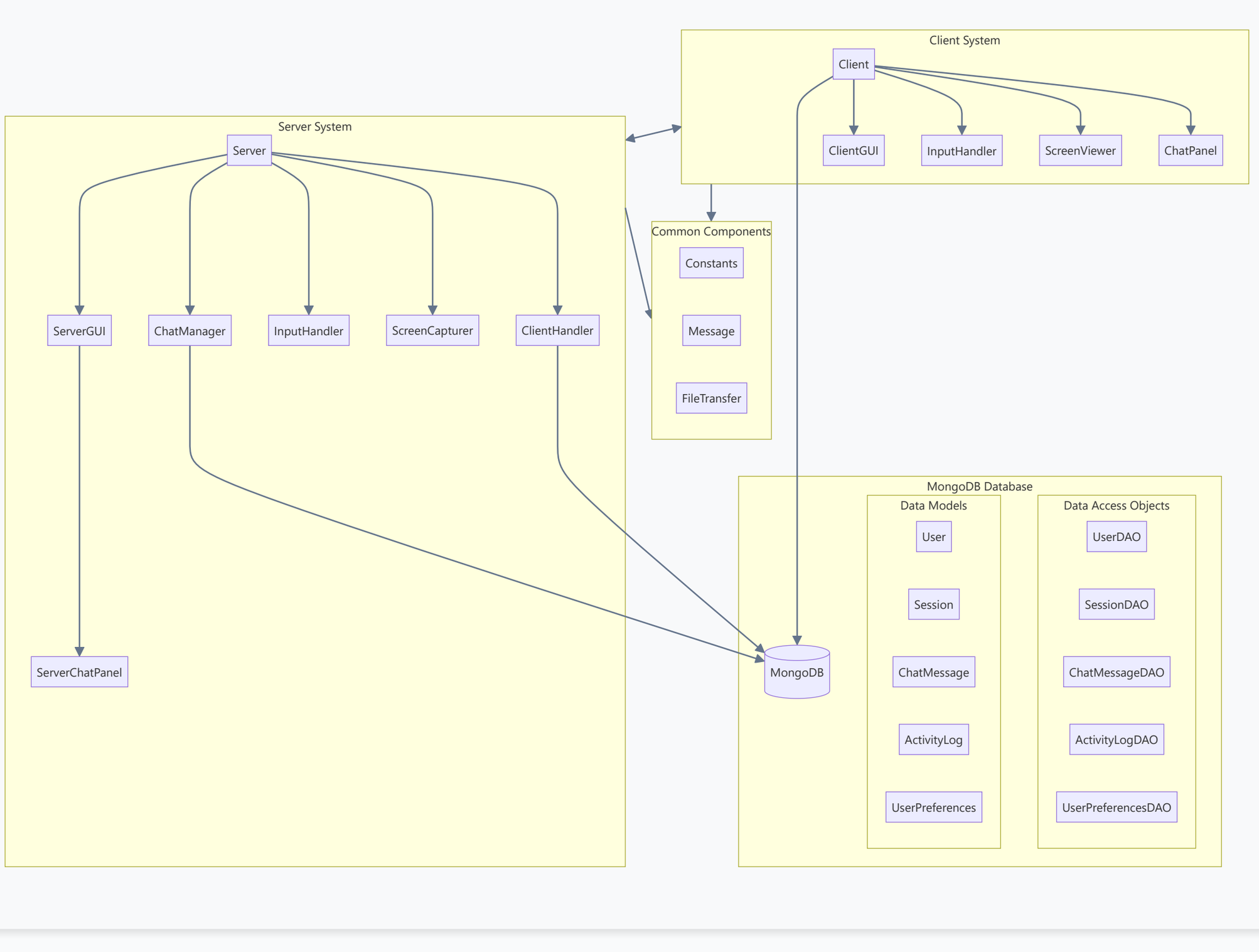
Allow clients to control the server's mouse and keyboard for remote assistance.

Chat & File Transfer

Built-in messaging system and file sharing capabilities for better collaboration.

System Architecture

High-Level Architecture



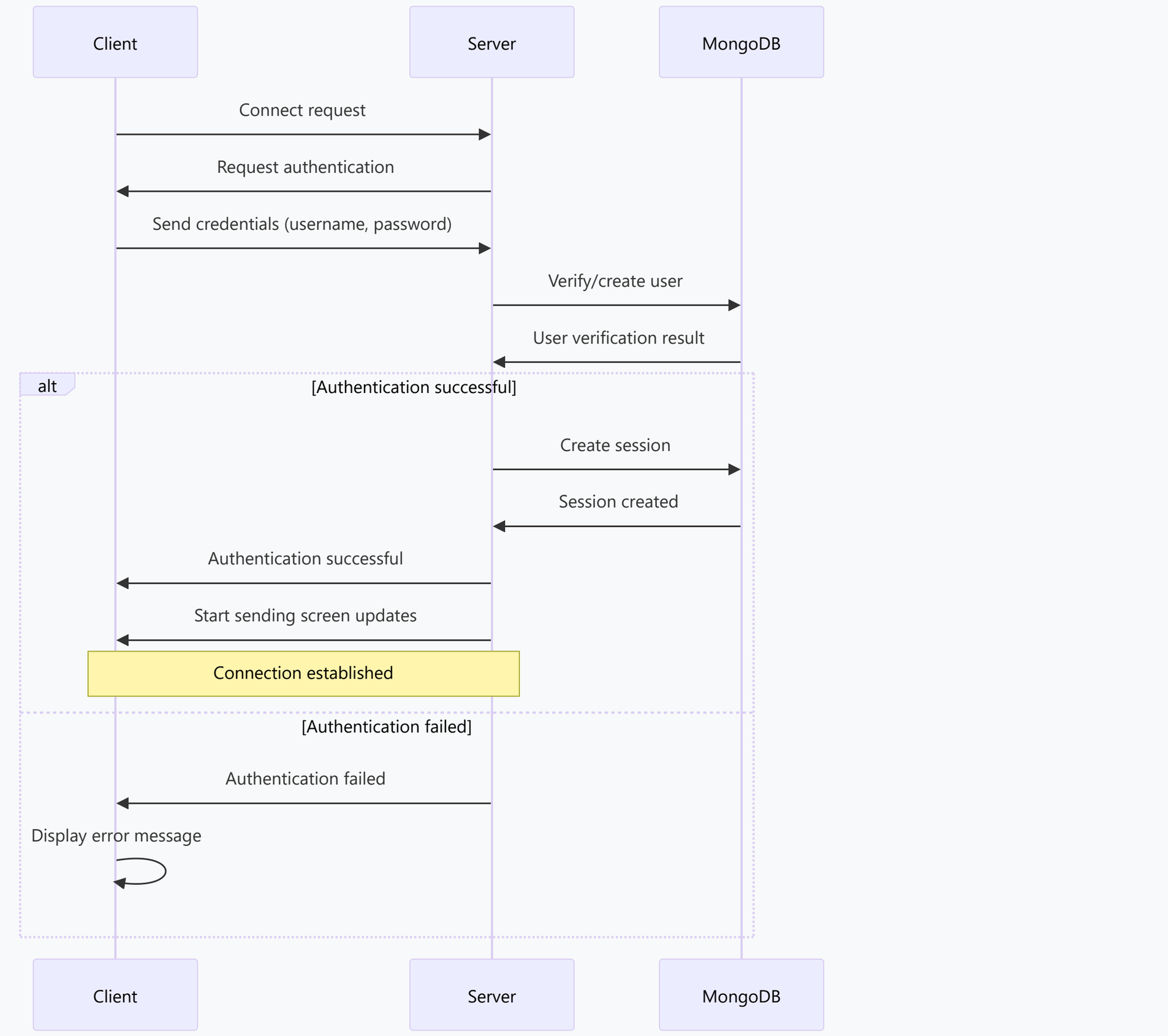
Architecture Components:

- **Client System:** Handles user interface, screen viewing, input capturing, and chat.
- **Server System:** Manages client connections, screen capturing, input processing, and chat distribution.
- **MongoDB Database:** Stores user data, session information, chat messages, and activity logs.
- **Common Components:** Shared utilities and data structures used by both client and server.

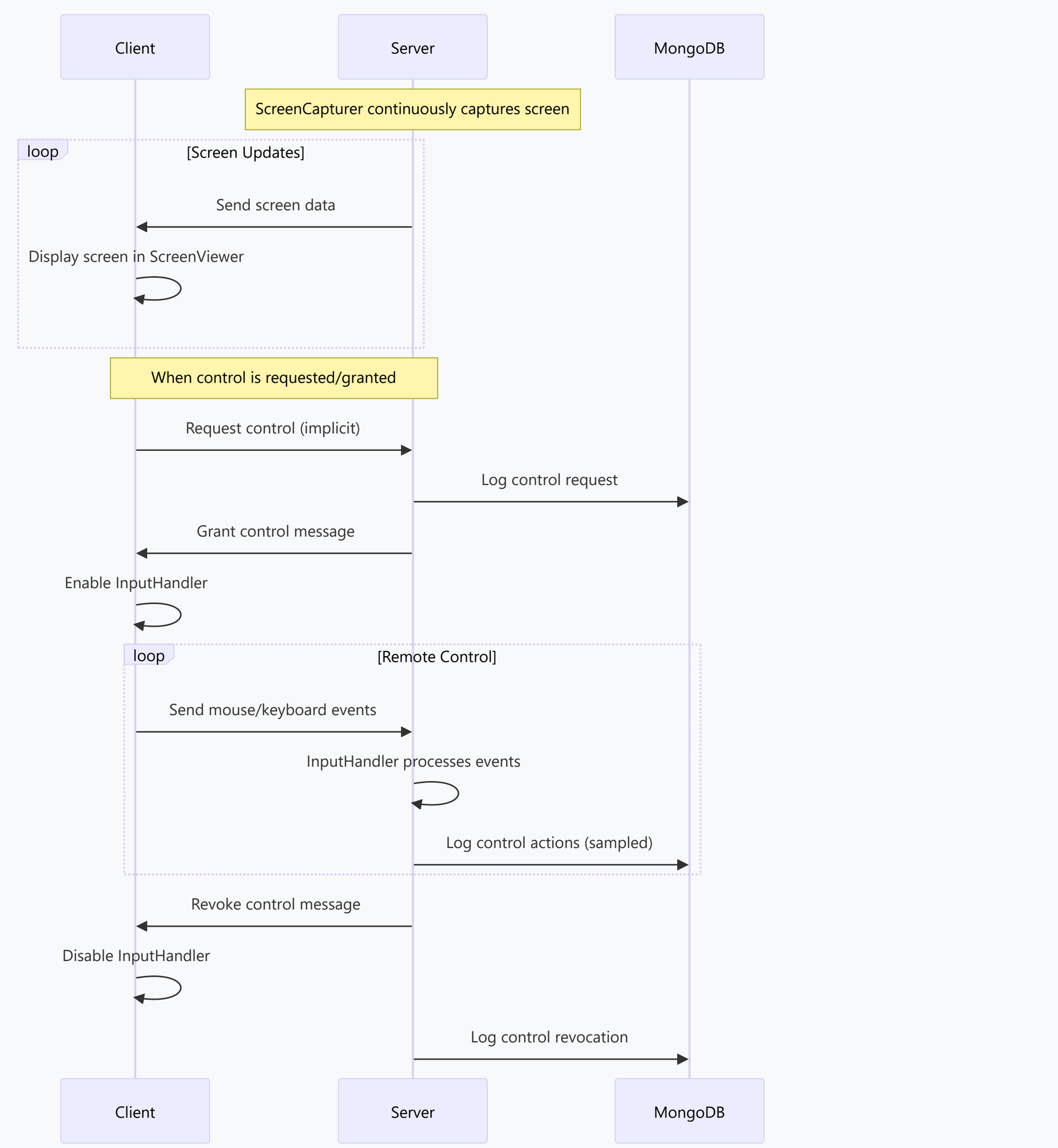
Directory Structure

```
remote_desktop_system/
└── src/
    ├── client/
    │   ├── ChatPanel.java        - Client-side chat interface
    │   ├── Client.java           - Core client functionality
    │   ├── ClientGUI.java        - Client user interface
    │   ├── InputHandler.java     - Handles mouse/keyboard input
    │   └── ScreenViewer.java     - Displays remote screen
    ├── common/
    │   ├── Constants.java        - Shared constants
    │   ├── FileTransfer.java     - File handling utilities
    │   ├── Message.java          - Message data structure
    │   ├── database/             - Database access layer
    │   │   ├── ActivityLogDAO.java
    │   │   ├── ChatMessageDAO.java
    │   │   ├── MongoDBConnection.java
    │   │   ├── SessionDAO.java
    │   │   ├── UserDAO.java
    │   │   └── UserPreferencesDAO.java
    │   └── model/                - Data models
    │       ├── ActivityLog.java
    │       ├── ChatMessage.java
    │       ├── Session.java
    │       ├── User.java
    │       └── UserPreferences.java
    └── server/
        ├── ChatManager.java      - Server-side chat handling
        ├── ClientHandler.java    - Manages client connections
        ├── InputHandler.java     - Processes remote input
        ├── ScreenCapturer.java   - Captures server screen
        ├── Server.java           - Core server functionality
        ├── ServerChatPanel.java  - Server-side chat interface
        └── ServerGUI.java        - Server user interface
```

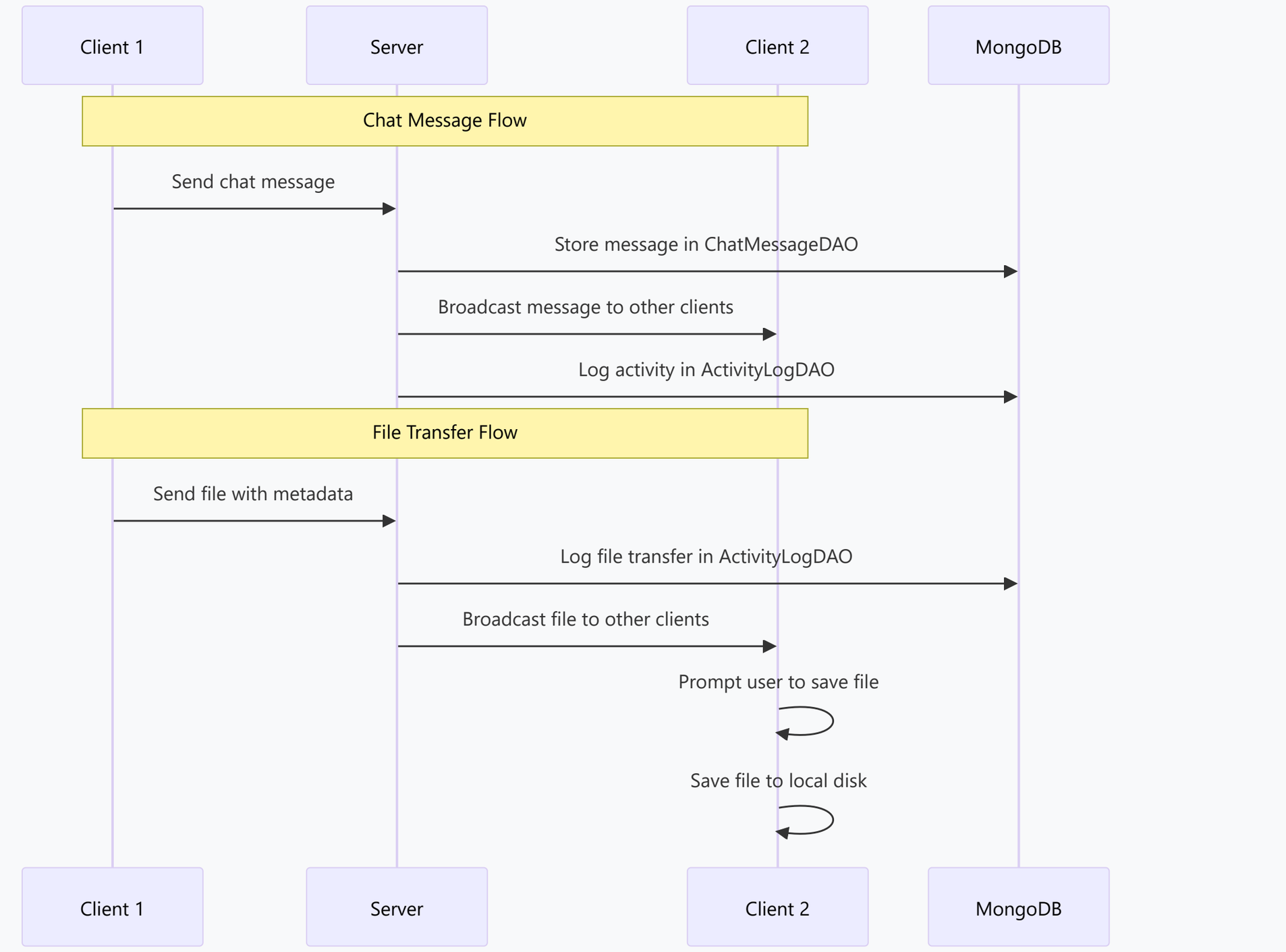
Connection & Authentication Flow



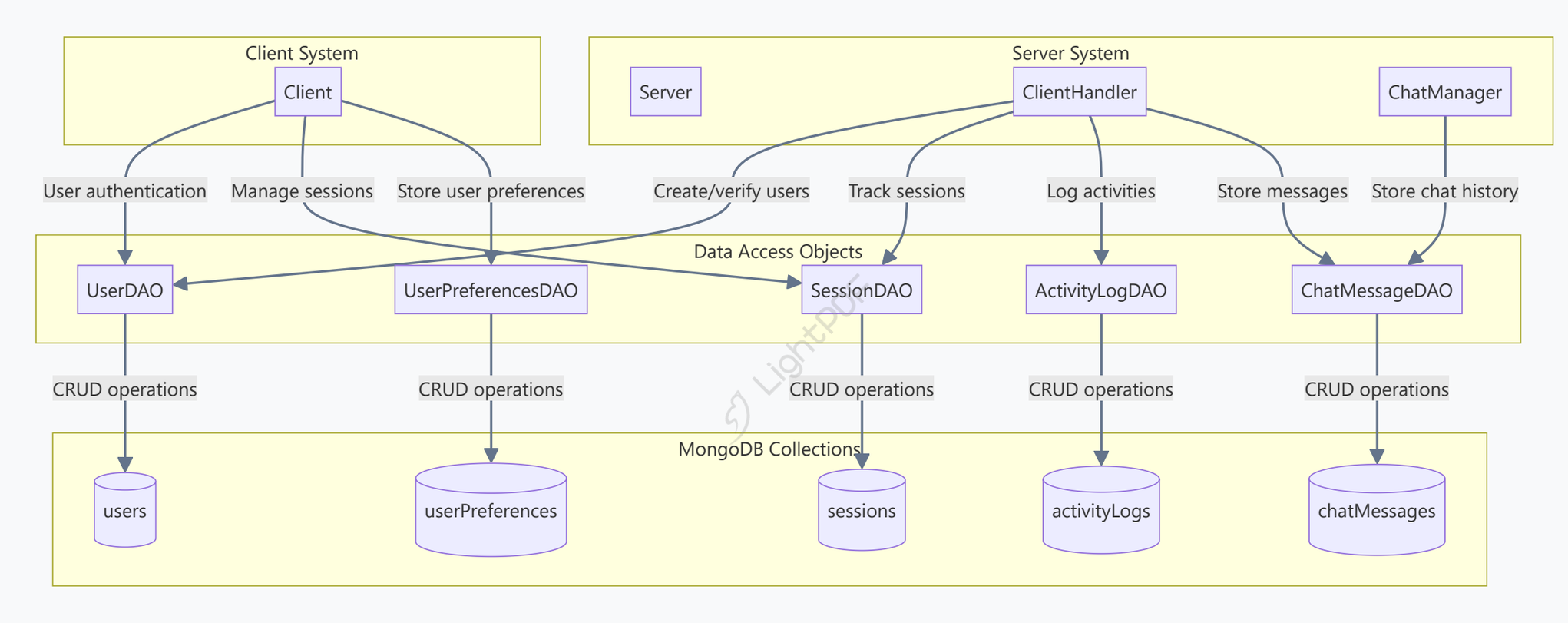
Screen Sharing & Remote Control Flow



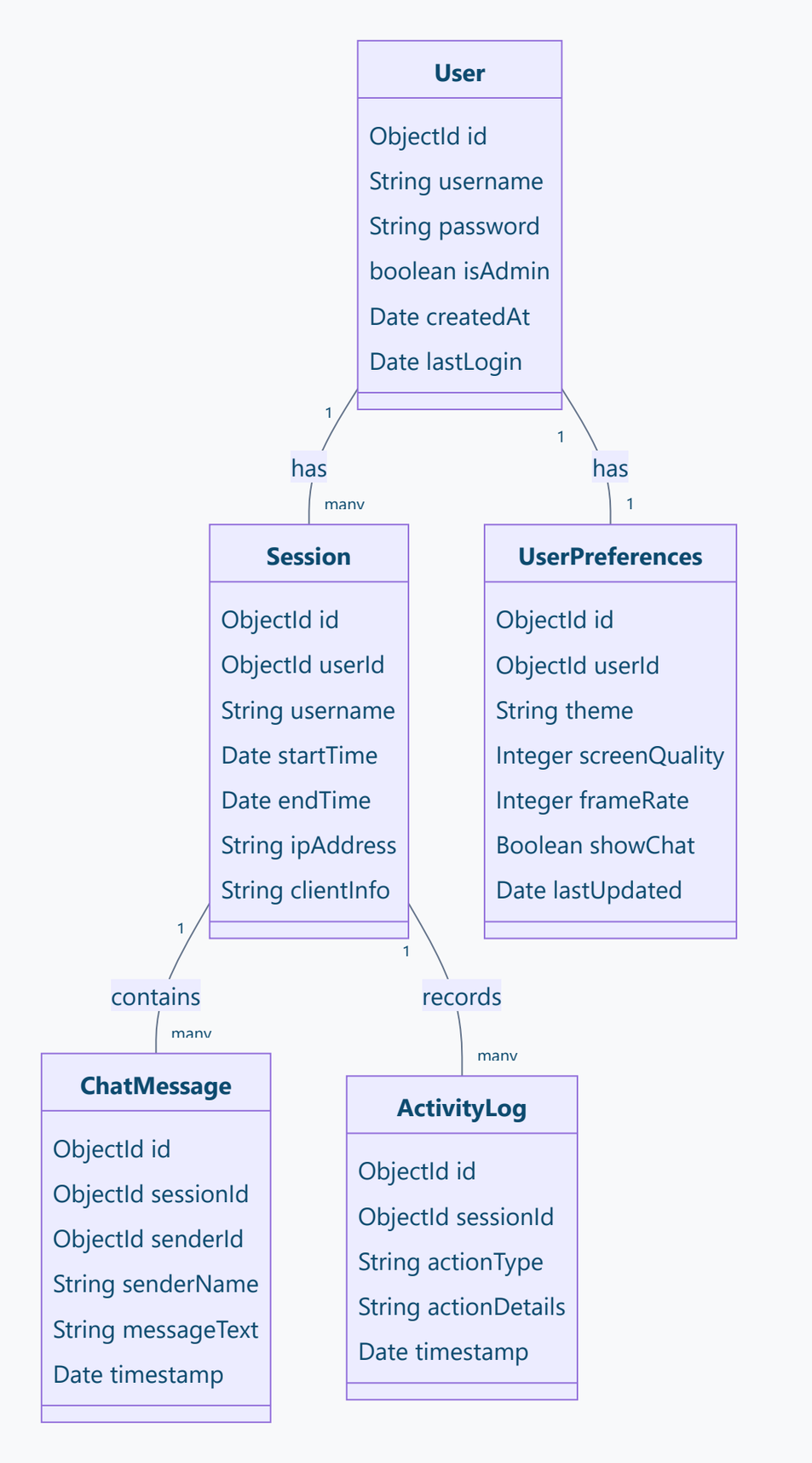
Chat & File Transfer Flow



Database Interaction Flow



Data Models & Relationships



Key Components Details

Client Side

Component	Description	Responsibilities
Client	Core client functionality	Manages server connection, authentication, message sending/receiving
ClientGUI	Client user interface	Provides UI for connection settings, chat, screen viewing
ScreenViewer	Remote screen display	Renders server's screen, handles image scaling and display options
InputHandler	Input event processor	Captures local mouse/keyboard events, sends to server
ChatPanel	Chat interface	Displays chat messages, allows sending messages and files

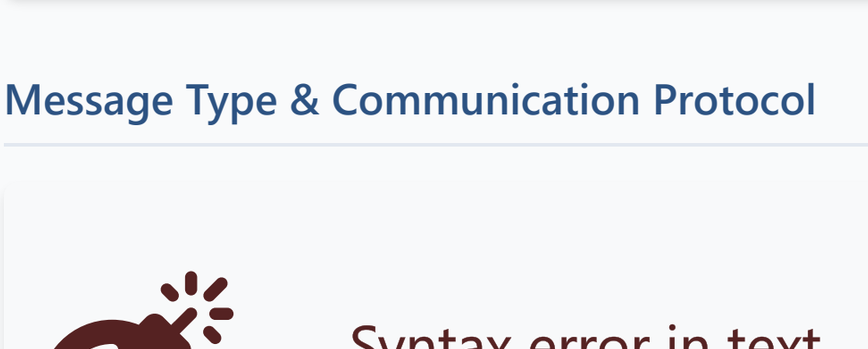
Server Side

Component	Description	Responsibilities
Server	Core server functionality	Listens for connections, manages clients, coordinates interactions
ClientHandler	Client connection manager	Handles individual client communication, authentication, message routing
ScreenCapturer	Screen capture utility	Captures server screen, compresses images, manages capture frame rate
InputHandler	Input processor	Processes remote input events, applies them to local system
ChatManager	Chat coordinator	Manages chat messages, stores history, broadcasts messages
ServerGUI	Server interface	Provides UI for server settings, client management, chat

Database Components

Component	Description	Responsibilities
UserDAO	User data access	User CRUD operations, authentication, password hashing
SessionDAO	Session data access	Tracks connection sessions, start/end times, client information
ChatMessageDAO	Chat message data access	Stores and retrieves chat messages, manages chat history
ActivityLogDAO	Activity log data access	Records system events, user actions, and control operations
UserPreferencesDAO	User preferences data access	Stores and retrieves user-specific settings and preferences

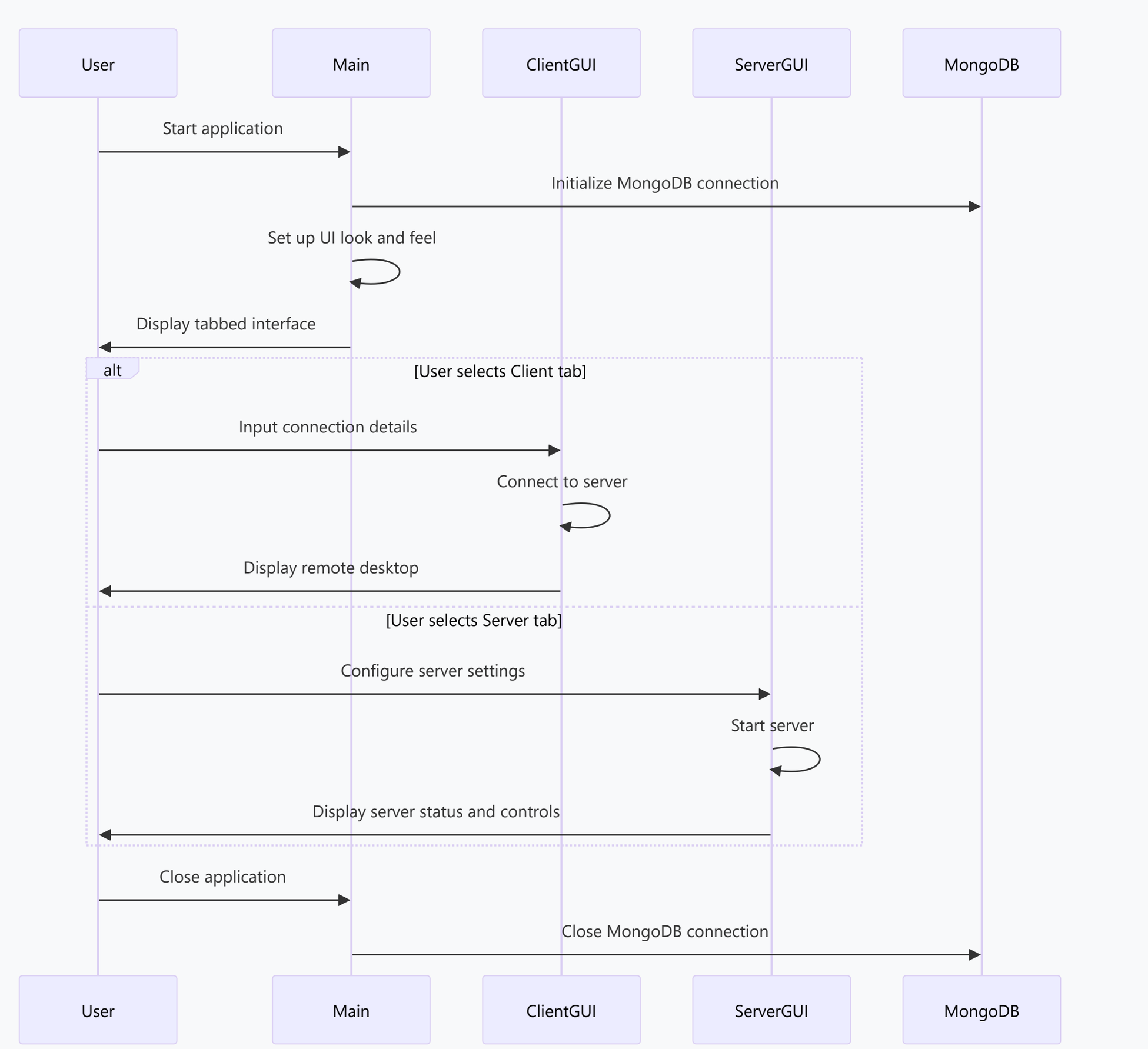
Message Type & Communication Protocol



The Remote Desktop System uses a binary protocol for communication between client and server. Each message starts with an integer message type, followed by type-specific data.

Message Type	Purpose	Data Format
AUTHENTICATION (0)	User login	password (String), username (String)
CHAT (1)	Text messages	sender (String), message (String)
FILE (2)	File transfer	sender (String), fileName (String), fileSize (int), fileData (byte[])
SCREEN (3)	Screen updates	dataSize (int), screenData (byte[])
MOUSE (4)	Mouse events	eventType (String), data-specific information (byte[])
KEYBOARD (5)	Keyboard events	eventType (String), keyCode (int), pressed (boolean)
CONTROL_GRANT (6)	Grant control	No data
CONTROL_REVOKE (7)	Revoke control	No data
DISCONNECT (8)	Client disconnect	No data

Main Application Flow



System Features & Capabilities

Screen Sharing Features

- Adjustable frame rate (15-120 FPS)
- Automatic performance tuning
- Configurable image quality
- Multi-monitor support
- Aspect ratio maintenance option
- Adjustable zoom for better visibility
- Fullscreen mode support

Remote Control Features

- Mouse movement tracking
- Mouse button handling (left, middle, right)
- Mouse wheel scrolling
- Keyboard input support
- Granular control permission management
- Coordinate mapping for different screen sizes
- Activity logging for security

Communication Features

- Real-time text chat
- Persistent chat history
- File transfer capabilities
- Automatic file saving options
- Message timestamps
- Server broadcast messages
- Chat participant tracking

Database & Logging Features

- User account management
- Secure password hashing
- Session tracking and statistics
- Activity logging for auditing
- User preference persistence
- Chat message history
- Performance metrics collection

System Scalability & Security

Scalability Considerations

- Connection pooling for database access
- Automatic resource management
- Dynamic frame rate adjustment based on system performance
- Image compression optimization
- Proper thread management for concurrent operations
- Efficient database indexing for faster queries

Security Features

- Password-based authentication
- Secure password storage with SHA-256 hashing
- Granular access control for remote system manipulation
- Comprehensive activity logging
- Session tracking and management
- Ability to revoke control at any time
- Connection validation and verification