**Project 2 Team: Arshia, Mahan, Saba, Paria, AmirMahdi**

**Breakdown of Classes and Responsibilities**

**1. Abstract Base Class: NetworkEntity**

- **Responsibilities:**
    - Define common attributes (_host, _port, _socket).
    - Provide common methods (_setup_socket, _accept_connection, _connect).
    - Abstract method start() which will be implemented by derived classes.
- **Purpose:** Acts as a foundation for both Server and Client classes, providing common functionality and enforcing a contract for derived classes.
- **Assigned to:** Mahan

**2. Server Class: Server**

- **Responsibilities:**
    - Inherit from NetworkEntity.
    - Implement the start() method to set up the server, accept connections, and handle client communication.
    - Implement _handle_client() to manage interactions with clients (receive guesses, validate them, and send responses).
- **Purpose:** Handles server-side operations, including managing connections and processing guesses.
- **Assigned to:** Arshia

**3. Client Class: Client**

- **Responsibilities:**
    - Inherit from NetworkEntity.
    - Implement the start() method to connect to the server, send guesses, and handle server responses.
- **Purpose:** Handles client-side operations, including sending guesses to the server and processing server responses.
- **Assigned to:** Saba

**4. Main Scripts: server.py and client.py**

- **Responsibilities:**
    - server.py: Initialize and start the Server class.

- client.py: Initialize and start the Client class, take user inputs for server host and port.

- **Purpose:** Serve as the entry points for running the server and client applications.
- **Assigned to:** Paria

**5. Communication Protocols and Error Handling**

- **Responsibilities:**
  - Design and implement communication protocols between the server and client, ensuring data consistency and error handling.
  - Implement custom exceptions for handling errors such as connection failures, timeouts, and invalid data.
  - Ensure that both Server and Client classes handle these protocols and exceptions properly.
- **Purpose:** Enhances the robustness and reliability of the communication between the server and client.
- **Assigned to:** AmirMahdi

**Responsibilities and Assignments**

**1. NetworkEntity (Assigned to Mahan)**

- **Tasks:**
  - Implement the NetworkEntity class in networking.py.
  - Define abstract methods and common socket operations.
  - Ensure proper encapsulation of socket operations.

**2. Server (Assigned to Arshia)**

- **Tasks:**
  - Implement the Server class in networking.py.
  - Implement the start() method to set up the server and listen for connections.
  - Implement _handle_client() to process client requests and send responses.
  - Ensure proper integration with the NetworkEntity base class.

**3. Client (Assigned to Saba)**

- **Tasks:**
  - Implement the Client class in networking.py.
  - Implement the start() method to connect to the server and handle user input.

- Manage sending guesses to the server and processing server responses.
- Ensure proper integration with the NetworkEntity base class.

**4. Main Scripts (server.py and client.py) (Assigned to Paria)**

- **Tasks:**
  - Create the server.py script to initialize and run the Server class.
  - Create the client.py script to initialize and run the Client class.
  - Ensure proper input handling for host and port.
  - Coordinate with Mahan, Arshia, and Saba to ensure scripts work with their respective classes.

**5. Communication Protocols and Error Handling (Assigned to AmirMahdi)**

- **Tasks:**
  - Design a robust communication protocol for the server-client interaction, ensuring reliable data transmission.
  - Implement custom exceptions for connection errors, data validation errors, and timeouts.
  - Ensure that these protocols and error-handling mechanisms are integrated into both the Server and Client classes.
  - Collaborate with Mahan, Arshia, and Saba to ensure that the communication protocol is consistently implemented.

**Detailed Relationships and Integration**

**1. NetworkEntity Class**

- **Abstract Class:** Provides the basic structure and common functionality.
- **Used By:** Server and Client classes to avoid code duplication and ensure consistency.

**2. Server Class**

- **Inherits:** NetworkEntity
- **Uses:** Methods from NetworkEntity for socket operations and connection handling.
- **Implements:** start() to manage server operations, _handle_client() for client-specific logic.

**3. Client Class**

- **Inherits:** NetworkEntity
- **Uses:** Methods from NetworkEntity for socket operations and connection management.

- **Implements:** start() to handle client operations and interactions with the server.

**4. Main Scripts**

- **server.py:** Initializes and starts the Server class, handles server-side execution.
- **client.py:** Initializes and starts the Client class, handles client-side execution.

**5. Communication Protocols and Error Handling**

- **Enhances:** Reliability and robustness of the Server and Client interactions.
- **Ensures:** Consistent handling of communication and errors across the system.