CISC 594 Executive Session 02

Client–Server Trivia Game System

Overview

In this assignment, you will design and implement a distributed client–server system that supports a multiple-choice trivia game. The system must allow multiple users to authenticate, play trivia questions by category, wager points on answers, and track persistent user statistics. The focus is not only on functionality, but also on clean architecture, protocol design, robustness, security, and extensibility.

This project simulates a real-world interactive networked application and should reflect professional software engineering practices.

System Description

The system consists of two main components:

1. Trivia Server
2. Trivia Client

The server hosts trivia questions, manages user accounts and game logic, and persists data. Clients connect to the server, authenticate users, request questions, submit answers, and display game state.

Functional Requirements

1. Server Requirements

The server is the authoritative source of truth and must support multiple concurrent clients.

1.1 User Management

The server must support:

- User registration (username + password)
- User login and logout
- Secure password storage (e.g., hashing + salting)
- Persistent storage of user data, including:
    - Total points
    - Games played
    - Correct / incorrect answers

    - Optional: per-category performance

The server must prevent:

- Duplicate usernames
- Unauthorized access to user data

1.2 Trivia Question Management

The server must:

- Store a collection of multiple-choice trivia questions
- Organize questions by category
- Ensure each question includes:
    - Category
    - Question text
    - Multiple answer options
    - Correct answer
- Select and send questions based on the client's requested category
- Avoid repeating recently used questions for the same user (reasonable strategy required)

Questions may be stored in:

- A database
- Structured files (e.g., JSON, XML)
- Any other justified persistent format

1.3 Game Logic

The server must:

- Receive a user's wager before sending a question
- Validate wagers (e.g., cannot exceed available points)
- Evaluate submitted answers
- Adjust user points based on correctness and wager amount
- Update and persist user statistics after each question
- Return the result of each round to the client

1.4 Communication

The server must:

- Expose a well-defined communication protocol (custom or standard)
- Handle malformed or unexpected client messages safely
- Support multiple clients simultaneously (threaded, async, or event-driven design)

2. Client Requirements

The client is the user-facing application.

2.1 Authentication Interface

The client must:

- Allow users to register and log in
- Provide clear feedback for authentication success or failure
- Maintain a session while connected

2.2 Gameplay Interface

The client must allow users to:

- Select a trivia category for the next question
- Choose a wager amount based on current points
- Receive and display the trivia question and answer options
- Submit an answer to the server
- View the result of their answer (correct/incorrect, points gained/lost)

The interface may be:

- Command-line based
- Graphical (desktop or web-based)

2.3 User Statistics

The client must be able to:

- Request and display current user statistics from the server
- Show at least:
    - Current point balance
    - Total questions answered
    - Accuracy or win/loss ratio

Testing Requirements

You must include:
Unit tests for core game logic
System tests for client–server communication

Deliverables

1. Source Code
    - Client

- Server

2. User Guide
    - How to run the server
    - How to run and use the client
3. Code and Unit Test Review Reports (at least 2 units)
    - Review Log
        - Issue ID
        - File/location
        - Severity
        - Category (code/test/design/process)
        - Resolution
    - Review Report
        - Review scope
        - Defect summary
        - Test quality assessment (e.g. coverage)
        - Review metrics
4. System Test Report
    - Description of testing strategy and results