Project 2 – QR Codes – Grading Sheet/Rubric

Evaluation?

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Grader: |  |  | Student Name: |  |  |  |
| Date/Time: |  |  | Student Name: |  |  |  |
| Team ID: |  |  | Student Name: |  |  |  |
| Late?: |  |  |  |  |  |  |
| Checkpoint?: |  |  |  | Project Score: |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Earned** | **Weight** | **Task ID** | **Description** |
| \_\_\_\_\_ | 20% | 1 | Basic client server communication. The client must establish a TCP socket connection with the server. The connection must be terminated gracefully. The code must be C/C++ and compiles correctly. |
| \_\_\_\_\_ | 20% | 2 | Simple message transmission. The client transmits a message and the server can respond to it. Example: a basic “echo server.” Prerequisite: Task 1. |
| \_\_\_\_\_ | 20% | 3 | Basic QR-code transmission/response. Client sends the size and the binary QR-code, the server receives it, decodes it, and replies with URL of the image, which the client receives. Prerequisite: Task 2. |
| \_\_\_\_\_ | 10% | 4 | Support for multiple clients, via processes or threads. Must properly address all concurrency issues that could arise. Prerequisite: Task 3. |
| \_\_\_\_\_ | 7.5% | 5 | Error checking (with security checks) and logging must be fully implemented and correct. Prerequisite: Task 3. |
| \_\_\_\_\_ | 7.5% | 6 | Proper support for rate limiting. Prerequisite: Task 3. |
| \_\_\_\_\_ | 5% | 7 | Command line arguments and dynamic listen port. Prerequisite: Task 3. |
| \_\_\_\_\_ | 5% | 8 | Proper support for time-outs. Prerequisite: Task 3. |
| \_\_\_\_\_ | 5% | 9 | Proper support for maximum users: Prerequisite: Task 4. |

Grader Notes: