**Experiment Report - 68 - test5\_demoCode**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | line 12 | Syntax | √ | × | × |
| E02 | line 45 | Semantic | × | × | × |
| E03 | line 57 | Logic | √ | √ | √ |

Additional Errors Found by Self: 0

Self-Review Detection Rate: 67%

Peer 1 Detection Rate: 33%

Peer 2 Detection Rate: 33%

1. **Source Code**
2. #include <string>
3. #include <fstream>
4. class TimeManager {
5. public:
6. TimeManager();
7. std::string getCurrentTimestamp();
8. void setTimeZone(const std::string& timeZone);
9. std::string getTimeZone() const;
10. private:
11. std::string timeZone;
12. };
13. class GPSModule {
14. public:
15. GPSModule();
16. std::string getCurrentLocation();
17. void updateGPSInfo();
18. private:
19. double latitude;
20. double longitude;
21. };
22. class VideoRecorder {
23. public:
24. VideoRecorder(TimeManager& timeManager, GPSModule& gpsModule, const VideoConfig& config);
26. void startRecording();
27. void stopRecording();
28. bool isRecording() const;
29. void embedMetadata(const std::string& metadata);
30. std::string getCurrentVideoFilePath() const;
31. private:
32. TimeManager& timeManager;
33. GPSModule& gpsModule;
34. VideoConfig videoConfig; // Configuration for the recording
35. std::ofstream videoFile;
36. std::string currentVideoFilePath;
37. bool recording;
38. void startRecording();
39. void writeMetadataHeader();
40. void embedTimestampAndGPS();
41. };
42. VideoRecorder::VideoRecorder(TimeManager& timeManager, GPSModule& gpsModule, const VideoConfig& config)
43. : timeManager(timeManager), gpsModule(gpsModule), videoConfig(config), recording(true) {}
44. VideoRecorder::~VideoRecorder() {
45. if (recording) {
46. stopRecording();
47. }
48. }