Assignment 2: Report

Group Members

- **Mohsen IranianGhareshiran** - 202382341
- **Galilea Le Moullec** - 202415993
- **Félicien Moquet** - 202415994
- **Kateryna Nazarenko** - 202415995

1 System Architecture and UML Diagrams

1.1 Class Diagram for Core Modules

The Automated Online Test Monitoring System consists of multiple modules:

- **User Module**: Handles test-taker authentication and access management.
- **Monitoring Module**: Utilizes AI-based facial recognition, gaze tracking, and motion detection.
- **Anti-Spoofing Module**: Implements deepfake detection and liveness verification.
- **Reporting Module**: Generates logs and alerts for test administrators.
- **Integration Module**: Ensures compatibility with existing LMS such as Moodle or ExamSoft.

1.2 Use Case Diagram for User Interactions

The following actors interact with the system:

- **Test-taker**: Participates in the online exam while being monitored.
- **Administrator**: Reviews reports and adjusts monitoring settings.
- **System AI**: Performs real-time analysis and reporting.

Use case interactions include:

- Start Exam (Test-taker initiates a session)
- Monitor Test (AI-driven real-time observation)

- Detect Fraudulent Behavior (Anti-spoofing mechanisms activate upon suspicious activity)
- Generate Report (System logs flagged behaviors and generates reports)
- Administrator Review (Admin evaluates reports and decides on any actions)

1.3 Sequence Diagram for Key Workflows

1. Test-taker logs in and starts the exam.
2. AI-driven monitoring begins (facial recognition, motion tracking, gaze detection).
3. If suspicious behavior is detected, an alert is generated.
4. The system logs the behavior and updates the administrator dashboard.
5. The administrator reviews the report post-exam.

2 Test Plan and Suite

2.1 Unit Tests

- **Facial Recognition Module**: Test AI accuracy in recognizing legitimate users.
- **Anti-Spoofing Measures**: Validate that deepfake detection correctly flags fraudulent activities.
- **Logging & Reporting**: Ensure flagged incidents are stored accurately in the database.

Tools: JUnit for backend logic, OpenCV for AI validation.

2.2 API Tests

If microservices are used, API tests will ensure:

- Secure authentication for test-takers.
- Efficient data exchange between LMS and monitoring system.
- Accurate data retrieval for post-exam reviews.

Tools: Postman, Jest (for API verification)

3 Justification of Design Choices

3.1 Transition to Microservices

Our system is structured using a **microservices architecture** to enable modularity, scalability, and ease of integration with existing LMS. The following design choices were made:

- **Decoupled AI Processing**: The AI-driven monitoring module operates independently, reducing system overhead.
- **Scalable Cloud Deployment**: The system is deployed on AWS/GCP, allowing horizontal scaling.
- **Secure Data Handling**: Only AI-generated reports are stored to maintain privacy compliance.

3.2 Why Microservices?

- **Modularity**: Easier debugging and independent module updates.
- **Scalability**: Allows handling thousands of concurrent test sessions.
- **Interoperability**: Seamless integration with various LMS platforms.

4 Team Contribution & Documentation

Team Member	**Contribution**
Mohsen IranianGhareshiran	Developed AI-based facial recognition and motion tracking
Galilea Le Moullec	Designed system architecture and microservices integration
Félicien Moquet	Worked on test plan, unit testing, and API verification
Kateryna Nazarenko	Developed reporting module and integration with LMS

4.1 Summary of Contributions

Each member contributed to both the conceptualization and implementation of the system. We collaborated on system design, ensuring alignment with the feasibility study.

5 Conclusion

This assignment outlines the core **design, architecture, and testing** framework for our **Automated Online Test Monitoring System**. By adopting **microservices architecture, AI-driven monitoring, and cloud deployment**, our system ensures **scalability, security, and reliability** in remote assessments. The proposed **testing strategy and design justification** further enhance the feasibility of a **cost-effective and privacy-conscious** solution for online proctoring.