Software Requirements Specification

for

Automated Exam Monitoring and Grading System

Version 1.0

Prepared by

Vaibhav Patidar(MSE2024009)

Aman Jaiswal(MSE2024010)

Pratik Pistom Rathod(MSE2024004)

Table of Contents

Revision History

Author	Date	Description	Version
Vaibhav patidar Pratik Rathod Aman jaiswal	04/05/2025	Initial Draft	1.0

1. Introduction

1.1 Purpose

The purpose of this document is to define the functional and non-functional requirements for the **Automated Exam Monitoring and Grading System**. This system is designed to streamline and digitize the process of administering, monitoring, and evaluating online examinations. It aims to maintain examination integrity through real-time monitoring powered by AI technologies and ensure efficient, accurate grading of both objective and subjective responses. The system targets academic institutions and professional certification bodies seeking a reliable and scalable exam solution.

1.2 Document Conventions

Throughout this document:

- The word **"shall"** indicates a mandatory requirement.
- The word **"should"** indicates a recommended or optional feature.
- Diagrams, where applicable, adhere to **UML (Unified Modeling Language)** standards.
- All timestamps are expressed in **24-hour format**.
- All textual content, data labels, and messages are written in **English**.

1.3 Intended Audience and Reading Suggestions

- **Developers**: Will use this document as a blueprint for system design and implementation.
- **Testers**: Will refer to this document to ensure the system conforms to the stated requirements.
- **Project Managers**: Will use the SRS for planning, milestone tracking, and requirement validation.
- **Clients and Stakeholders**: Will use the SRS to understand the capabilities and scope of the system, ensuring it aligns with institutional needs.

1.4 Product Scope

This system is a comprehensive web-based platform for conducting online exams. It supports secure login, real-time monitoring through webcam and screen sharing, and uses AI to detect suspicious behavior such as face absence or presence of additional individuals. Objective-type questions are automatically graded, while subjective responses are routed to human evaluators for review. After assessment, the system generates detailed performance reports. It is scalable for large institutions and customizable for various exam formats.

1.5 References

- IEEE Std 830-1998 – Recommended Practice for Software Requirements Specifications.
- UML User Guide by Booch, Rumbaugh, and Jacobson.
- Supporting diagrams (use case, class, sequence diagrams) are included in **Appendix B**.

---

2. Overall Description

2.1 Product Perspective

The Automated Exam Monitoring and Grading System is a standalone web application that interfaces with:

- Authentication systems for secure user login.
- AI-powered video and screen analysis modules for exam surveillance.
- An evaluation engine that supports both automatic and manual grading workflows.

It includes four main modules:

- **Student Interface**: For taking exams, viewing instructions, and submitting responses.
- **Invigilator Dashboard**: For live proctoring and flag review.
- **Evaluation Engine**: For automated and manual grading.
- **Monitoring Module**: Uses AI to flag behavior that may violate exam rules.

2.2 Product Functions

The core functions of the system include:

- Managing the complete exam lifecycle from scheduling to result generation.
- Facilitating secure student authentication and access to exams.
- Monitoring student activity via live webcam and screen capture.
- Identifying and flagging suspicious behaviors using AI algorithms.
- Auto-grading objective questions based on a predefined answer key.
- Routing subjective answers to evaluators with annotation tools.
- Generating exam reports and performance summaries.

2.3 User Classes and Characteristics

- **Admin**: Oversees the entire system, configures settings, schedules exams, and manages users.
- **Invigilator**: Monitors live video feeds and reviews AI-generated flags.
- **Student**: Logs in to take the exam and submits responses; requires webcam and mic access.
- **Evaluator**: Grades subjective responses, provides feedback, and finalizes scores.

Each user class has a dedicated dashboard tailored to their roles and responsibilities.

2.4 Operating Environment

The system will operate on modern web infrastructure and is compatible with:

- **Web Browsers**: Google Chrome and Mozilla Firefox (latest versions).
- **Hardware**: Devices with a webcam, microphone, and minimum 4 GB of RAM.
- **Servers**: Linux or Windows servers for backend processing.
- **Databases**: MySQL or PostgreSQL.

- **Frameworks**: Backend may be built on Django (Python) or Node.js (JavaScript).

2.5 Design and Implementation Constraints

- The system must conform to the academic institution's policies on exams and privacy.
- Monitoring data must be processed and flagged in real-time with minimal delay.
- AI-based detection should be non-intrusive, ensuring the exam experience is uninterrupted.

2.6 User Documentation

The following documents will support the deployment and use of the system:

- **User Manual**: Includes tutorials for students, evaluators, and admins.
- **Installation Guide**: For setting up the system on institutional infrastructure.
- **Troubleshooting Guide**: FAQs and solutions to common issues.

2.7 Assumptions and Dependencies

- Users will access the system using stable internet connections.
- Devices used by students and invigilators are equipped with a functional webcam and microphone.
- Integration with Learning Management Systems (LMS) like Moodle or Blackboard is optional but supported.

---

3. External Interface Requirements

3.1 User Interfaces

- **Login/Signup Page**: Secure authentication for different user roles.

- **Student Dashboard**: Displays upcoming exams, instructions, and live countdown timer.
- **Invigilator Panel**: Shows real-time video streams, flagged behavior alerts, and session summaries.
- **Evaluator Interface**: Displays submitted responses, grading tools, and feedback options.
- **Admin Panel**: Exam creation, user management, and report generation tools.

3.2 Hardware Interfaces

- **Webcam**: Required for capturing the candidate's facial activity.
- **Microphone**: Optionally used for audio-based flagging.
- **System Memory**: Devices must have at least 4 GB of RAM to ensure smooth performance.

3.3 Software Interfaces

- Integration with:
    - **AI-based detection APIs** for flagging irregular behavior.
    - **Database** systems (MySQL/PostgreSQL) for data storage.
    - **Authentication servers** using OAuth2 or SAML protocols.

3.4 Communications Interfaces

- The application uses **HTTP/HTTPS** for data transmission.
- **WebSockets** are employed for real-time video and screen sharing.
- **REST APIs** support backend operations such as data fetching and exam submission.

---

4. System Features

4.1 Feature: Take Online Exam

Students can securely log in, access exam instructions, and begin answering questions. Features include:

- Auto-saving responses every few seconds with timestamp logs.
- A visible countdown timer and automatic submission once time expires.
- Navigational controls for moving between questions.

4.2 Feature: Monitor Exam Session

Invigilators oversee exam activity using a monitoring dashboard. Key components include:

- Live webcam and screen streaming.
- Real-time alerts for suspicious activity detected by AI (e.g., missing face, multiple faces, tab switching).
- A log of all flagged events for post-exam review.

4.3 Feature: Evaluate and Grade Exams

Evaluation is divided into two parts:

- **Objective Questions**: Instantly graded based on an answer key.
- **Subjective Questions**: Presented to human evaluators with space for scores and comments.
  Evaluators can override auto-grades, leave feedback, and finalize marks for report generation.

---

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system should seamlessly support **100 or more concurrent users**.

- AI monitoring tools should detect and flag activity within **2 seconds** of its occurrence.

5.2 Safety Requirements

- Student progress must be saved **periodically** (e.g., every 30 seconds).
- System should gracefully handle **loss of webcam/mic** without data loss.

5.3 Security Requirements

- All data transfers must be secured with **SSL encryption**.
- **Role-based access control (RBAC)** will prevent unauthorized access.
- **Facial recognition** will be used to verify student identity before and during the exam.

5.4 Software Quality Attributes

- **Availability**: System uptime should be **99.5% or higher**.
- **Usability**: The interface must be intuitive and accessible for users of all technical levels.
- **Maintainability**: Modular code architecture to simplify debugging and feature updates.
- **Reliability**: High accuracy of AI flagging and grading algorithms is crucial.

5.5 Business Rules

- Exams are locked and begin **precisely at the scheduled time**.
- Any flagged sessions must be reviewed by an invigilator before results are published.
- Final grades require **evaluator confirmation**, especially for subjective sections.

---

6. Other Requirements

- Video and activity logs must be stored securely for a **minimum of 6 months**.
- After each exam, students are prompted to provide **feedback**.
- System must comply with **accessibility guidelines** (such as **WAI-ARIA**) to support users with disabilities.

Appendices

Appendix A: Glossary

- **Flagged Behavior**: Any suspicious activity during the exam, such as multiple faces or switching tabs.
- **Evaluator**: An authorized person responsible for scoring subjective questions.
- **Auto-grading**: The automatic scoring mechanism for objective questions using predefined keys.

Appendix B: Analysis Models

- **Use Case Diagrams**: Visual representation of how users interact with the system.
- **Class Diagrams**: Illustrates the object-oriented structure and relationships between entities.
- **Sequence Diagrams (optional)**: Describes the flow of operations over time between components.

Appendix C: To Be Determined (TBD)

- Final **grading rubrics** for subjective responses.
- Complete **UI/UX design** for all dashboards and panels.
- Optional **integration** strategy with institutional LMS.