



Namal University, Mianwali

Department of Computer Science

System Requirement Specification For Fair and Unbiased Assessment System

Course Code	CSC-225
Course Name	Software Engineering

Team Members		
Name	Roll No	Emails
Abdul Rehman	01	bscs24f01@namal.edu.pk
Ahmad Ali	02	bscs24f02@namal.edu.pk
Junaid Gondal	28	bscs24f28@namal.edu.pk

Submission Date: December 29, 2025

Contents

1	Introduction	5
1.1	Purpose	5
1.1.1	Purpose of SRS	5
1.1.2	Intended Audience	5
1.1.2.1	Stakeholders	5
1.1.2.2	Development Teams	5
1.1.2.3	Testers	5
1.1.2.4	End Users	6
1.2	Scope	6
1.2.1	Software Product Identification	6
1.2.2	Product Functions	6
1.2.3	Application Description	7
1.2.3.1	Benefits, Objectives, and Goals	7
1.2.4	Consistency with Higher-Level Specifications	7
1.3	Definitions, Acronyms, and Abbreviations	7
1.4	References	7
1.4.1	Standards	7
1.4.2	Technical Documentation	8
1.4.3	Development Tools	8
1.5	Overview	8
1.5.1	Document Structure	8
1.5.2	Organization of the Document	8
2	General Description	9
2.1	Product Perspective	9
2.1.1	System Context	9
2.1.2	User Interfaces	9
2.1.3	Hardware Interfaces	9
2.1.4	Software Interfaces	9
2.1.5	Communications Interfaces	10
2.2	Product Functions	10
2.2.1	Data Capture and Digitization	10
2.2.2	Assessment Evaluation and Scoring	10
2.2.3	Reporting and Analytics	10
2.2.4	Course Management	10
2.3	User Characteristics	10
2.3.1	Students	10
2.3.2	Teachers	11
2.3.3	Academic Administrators	11

2.4	General Constraints	11
2.4.1	Regulatory and Policy Constraints	11
2.4.2	Hardware Limitations	11
2.4.3	Interfaces to External Applications	11
2.4.4	Parallel Operation Constraints	11
2.4.5	Logging Constraints	11
2.4.6	Control and Administration Constraints	11
2.4.7	Safety and Security Considerations	11
2.5	Assumptions and Dependencies	12
2.5.1	Handwriting Legibility Assumption	12
2.5.2	External AI System Dependency	12
3	Specific Requirements	12
3.1	Functional Requirements	12
3.1.1	Student Login with Given Credentials	12
3.1.1.1	Introduction	12
3.1.1.2	Inputs	12
3.1.1.3	Processing	12
3.1.1.4	Outputs	12
3.1.1.5	Performance Requirements	13
3.1.1.6	Design Constraints	13
3.1.1.7	Attributes	13
3.1.1.8	Other Requirements	13
3.1.2	Uploading Quiz and Assignment Answer Images	13
3.1.2.1	Introduction	13
3.1.2.2	Inputs	14
3.1.2.3	Processing	14
3.1.2.4	Outputs	14
3.1.2.5	Performance Requirements	14
3.1.2.6	Design Constraints	14
3.1.2.7	Attributes	15
3.1.2.8	Other Requirements	15
3.1.3	Viewing Quiz/Assignment Report and Class Performance Graph	15
3.1.3.1	Introduction	15
3.1.3.2	Inputs	15
3.1.3.3	Processing	15
3.1.3.4	Outputs	16
3.1.3.5	Performance Requirements	16
3.1.3.6	Design Constraints	16
3.1.3.7	Attributes	16

3.1.3.8	Other Requirements	16
3.1.4	Administrator Assign Courses and Create Faculty Credentials . .	17
3.1.4.1	Introduction	17
3.1.4.2	Inputs	17
3.1.4.3	Processing	17
3.1.4.4	Outputs	17
3.1.4.5	Performance Requirements	17
3.1.4.6	Design Constraints	17
3.1.4.7	Attributes	17
3.1.4.8	Other Requirements	18
3.1.5	Administrator Enroll Students and Create Credentials	18
3.1.5.1	Introduction	18
3.1.5.2	Inputs	18
3.1.5.3	Processing	18
3.1.5.4	Outputs	18
3.1.5.5	Performance Requirements	18
3.1.5.6	Design Constraints	18
3.1.5.7	Attributes	19
3.1.5.8	Other Requirements	19
3.1.6	Administrator Credential Update	19
3.1.6.1	Introduction	19
3.1.6.2	Inputs	19
3.1.6.3	Processing	19
3.1.6.4	Outputs	19
3.1.6.5	Performance Requirements	20
3.1.6.6	Design Constraints	20
3.1.6.7	Attributes	20
3.1.6.8	Other Requirements	20
3.1.7	Teacher Login and Dashboard	20
3.1.7.1	Introduction	20
3.1.7.2	Inputs	20
3.1.7.3	Processing	20
3.1.7.4	Outputs	20
3.1.7.5	Performance Requirements	20
3.1.7.6	Design Constraints	21
3.1.7.7	Attributes	21
3.1.7.8	Other Requirements	21
3.1.8	Teacher Create Assessment	21
3.1.8.1	Introduction	21
3.1.8.2	Inputs	21

3.1.8.3	Processing	21
3.1.8.4	Outputs	21
3.1.8.5	Performance Requirements	21
3.1.8.6	Design Constraints	21
3.1.8.7	Attributes	22
3.1.8.8	Other Requirements	22
3.1.9	Automated Assessment Checking and Report Generation	22
3.1.9.1	Introduction	22
3.1.9.2	Inputs	22
3.1.9.3	Processing	22
3.1.9.4	Outputs	22
3.1.9.5	Performance Requirements	22
3.1.9.6	Design Constraints	22
3.1.9.7	Attributes	22
3.1.9.8	Other Requirements	22
3.1.10	Teacher Control Over Student Marks Visibility	23
3.1.10.1	Introduction	23
3.1.10.2	Inputs	23
3.1.10.3	Processing	23
3.1.10.4	Outputs	23
3.1.10.5	Performance Requirements	23
3.1.10.6	Design Constraints	23
3.1.10.7	Attributes	23
3.1.10.8	Other Requirements	23
3.2	External Interface Requirements	23
3.2.1	User Interfaces	23
3.2.1.1	Performance Requirements	24
3.2.1.2	Design Constraints	24
3.2.1.3	Attributes	25
3.2.1.4	Other Requirements	25
3.2.1.5	Hardware Interfaces	26
3.2.1.6	Software Interfaces	26
3.2.1.7	Communications Interfaces	27
4	Appendix	27
4.1	Use Case Diagram	27
4.2	Context Diagram	27

1 Introduction

1.1 Purpose

1.1.1 Purpose of SRS

The primary purpose of this Software Requirements Specification (SRS) is to present a complete and clear description of the **Fair and Unbiased Assessment System (FUAS)**. This document provides detailed information about the system's functional and non-functional requirements, enabling stakeholders to understand what the system is expected to deliver.

The SRS serves as a shared reference for stakeholders, developers, and testers. It ensures that developers implement the system according to specified requirements and that testers validate all functionalities as demanded by stakeholders. Overall, this document acts as a communication bridge among all team members and stakeholders.

1.1.2 Intended Audience

This Software Requirements Specification (SRS) is intended for stakeholders, developers, end users, and testers to ensure that all parties are well informed about the Fair and Unbiased Assessment System.

1.1.2.1 Stakeholders

- **Institutional Decision Makers:** Ensure that the system aligns with educational objectives and regulatory requirements such as fairness, accountability, scalability, copyright compliance, and transparency.
- **Project Managers:** Monitor project progress, ensure resource availability, and verify that project goals are achieved within the defined schedule.

1.1.2.2 Development Teams

- **Software Engineers:** Use this SRS as a roadmap to implement functional and non-functional requirements and develop a reliable, fair, and unbiased assessment system.
- **System Architects:** Design the system's high-level architecture to ensure scalability, reliability, integration, proper data flow, and fault tolerance.

1.1.2.3 Testers

- **Quality Assurance Specialists:** Design and execute test cases to ensure that all functional requirements operate correctly and that the assessment process remains

fair and accurate.

1.1.2.4 End Users

- **Assessors and Candidates:** Verify that the system provides unbiased evaluation, transparency, and a positive user experience.

1.2 Scope

1.2.1 Software Product Identification

The software product is named **Fair and Unbiased Assessment System (FUAS)**. The system consists of the following major components:

- Registration and authentication
- Evaluation and scoring
- Report generation module
- Content management module
- System administration

1.2.2 Product Functions

The Fair and Unbiased Assessment System is designed to:

- Register users and verify user credentials
- Assign courses to teachers and students
- Allow students to upload handwritten answer images
- Enable creation of sections for quizzes and tests
- Allow teachers to upload answer keys
- Automatically evaluate quizzes and tests based on answer keys and critical thinking analysis
- Prevent modification of marks once uploaded
- Generate assessment reports
- Ensure data integrity and privacy

The system will not:

- Automatically generate answer keys
- Replace or modify any submitted quiz or test

1.2.3 Application Description

The Fair and Unbiased Assessment System is designed for use in educational institutions to evaluate assignments, quizzes, and tests without bias or favoritism. The system ensures that all students are treated equally and that marks are awarded strictly based on performance rather than personal influence.

1.2.3.1 Benefits, Objectives, and Goals

- **Fairness and Bias Reduction:** Ensures assessments are marked solely on performance.
- **Transparency:** Provides a clear and uniform evaluation process accessible to authorized stakeholders.
- **Accuracy and Reliability:** Delivers objective assessment accuracy of more than 99%.
- **Scalability:** Supports up to 10,000 candidates without performance degradation.
- **Efficiency:** Evaluates assessments and generates reports within five minutes.
- **Security:** Protects user data through encryption and role-based access control.

1.2.4 Consistency with Higher-Level Specifications

This SRS is consistent with higher-level system requirement specifications that emphasize fairness, equality, transparency, and compliance with educational and regulatory standards.

1.3 Definitions, Acronyms, and Abbreviations

This section defines the terms, acronyms, and abbreviations used throughout this document.

1.4 References

1.4.1 Standards

1. IEEE Std 830-1984: IEEE Recommended Practice for Software Requirements Specifications.
2. IEEE Std 1002-1987: IEEE Standard Taxonomy for Software Engineering Standards.

Term/Acronym	Definition
FUAS	Fair and Unbiased Assessment System
SRS	Software Requirements Specification
UI/UX	User Interface / User Experience
Assessment	A structured evaluation of knowledge or skills
Candidate	An individual registered for assessment
Assessor	Authorized person to manage assessments
Bias Detection	Mechanisms to reduce unfair evaluation
Evaluation Engine	Automated scoring component
Audit Log	Secure record of system activities
QA	Quality Assurance
RBAC	Role-Based Access Control
API	Application Programming Interface
JSON	JavaScript Object Notation

Table 1: Definitions, Acronyms, and Abbreviations

1.4.2 Technical Documentation

1. MongoDB Documentation: <https://www.mongodb.com/docs>
2. Express.js Documentation: <https://expressjs.com>
3. Node.js Documentation: <https://nodejs.org>

1.4.3 Development Tools

1. Visual Studio Code: <https://code.visualstudio.com>
2. GitHub Guide: <https://docs.github.com>

1.5 Overview

1.5.1 Document Structure

This SRS document is organized as follows:

1. Introduction
2. Overall Description
3. Specific Requirements
4. Appendices and References

1.5.2 Organization of the Document

- Sections are numbered hierarchically for clarity.
- Each section focuses on a specific system aspect, with subsections providing detailed explanations.

2 General Description

2.1 Product Perspective

The Fair and Unbiased Assessment System (FUAS) addresses key concerns of both students and instructors. Students often perceive bias in grading, which can erode trust in instructors. Conversely, instructors may face unfounded accusations of favoritism. FUAS resolves this trust issue by providing unbiased, AI-assisted evaluation of quizzes and assignments. The system can process handwritten assessments uploaded by students, evaluate them based on provided answer keys and logical reasoning, and generate detailed reports highlighting errors.

The system will be available as both a mobile application and a web-based platform, each serving the same functional role.

2.1.1 System Context

The system interacts with three primary entities: students, instructors, and administrators. Each entity has its own interface and functionality according to their role.

2.1.2 User Interfaces

- **Mobile Interface:** Includes all features of the web interface plus an option to capture and upload images directly using the device camera.
- **Web Interface:** Provides all functionalities except for the direct image capture feature available on mobile.

2.1.3 Hardware Interfaces

- **Computing Devices:** Smartphones, laptops, desktops, and tablets used to interact with the system.
- **Server / Web Host:** Requires sufficient processing capacity and memory to handle concurrent image uploads and assessment data.

2.1.4 Software Interfaces

- **Mark Assist Engine:** An AI-based evaluation component that grades assessments and highlights mistakes based on logical and critical thinking.
- **Database System:** Stores uploaded images, extracted text, assessment metadata, and evaluation results securely.

2.1.5 Communications Interfaces

- Secure network protocols manage communication between mobile and web frontends and backend services.
- Standard web communication technologies support image uploads and retrieval of evaluation results.

2.2 Product Functions

The primary functions of the system focus on what the system does rather than how it does it, which is detailed in the Functional Requirements section.

2.2.1 Data Capture and Digitization

The system accepts images of assessments in PDF or image formats. The AI tool extracts text from handwritten submissions for evaluation.

2.2.2 Assessment Evaluation and Scoring

The AI engine evaluates student responses against teacher-provided answer keys while considering critical thinking. Assessments are graded fairly and without bias.

2.2.3 Reporting and Analytics

The system automatically generates reports for students and instructors. Students view their own reports, while instructors access class-level analytics and summaries.

2.2.4 Course Management

Courses are assigned to instructors and students. The system supports different courses for different years and maintains appropriate course-instructor-student relationships.

2.3 User Characteristics

The system is designed specifically for educational use by students, instructors, and academic administrators, with functionality tailored to each role.

2.3.1 Students

Students upload images of assessments, receive reports with scores and feedback, and identify mistakes through the system-generated analysis.

2.3.2 Teachers

Teachers upload answer keys, and the system evaluates student submissions automatically. Reports show top, average, and lowest scores, as well as overall class performance.

2.3.3 Academic Administrators

Administrators manage course assignments, maintain student and instructor records, and assist users in case of credential loss or access issues.

2.4 General Constraints

2.4.1 Regulatory and Policy Constraints

All student assessment data must comply with institutional and educational data protection policies, including access control, auditing, and retention rules.

2.4.2 Hardware Limitations

The system relies on mobile devices for image capture and servers for processing. Low-resolution images may affect assessment evaluation accuracy.

2.4.3 Interfaces to External Applications

FUAS does not rely on external applications; the AI evaluation tool is integrated within the system.

2.4.4 Parallel Operation Constraints

Multiple users may interact with the system simultaneously, including peak usage periods during assessment submissions.

2.4.5 Logging Constraints

System activities must be recorded to ensure traceability, accountability, and dispute resolution.

2.4.6 Control and Administration Constraints

Administrative tasks such as user management, course creation, and assessment configuration require role-based access control to maintain system integrity.

2.4.7 Safety and Security Considerations

All academic data must be protected from unauthorized access, misuse, and loss using encryption, secure storage, and controlled access.

2.5 Assumptions and Dependencies

2.5.1 Handwriting Legibility Assumption

Student handwriting and image quality must meet a minimum standard for the AI tool to accurately process and evaluate assessments. Guidelines are provided to ensure proper image capture and legibility.

2.5.2 External AI System Dependency

The AI evaluation engine must be available and operational to process and evaluate assessments. Without it, the automated assessment functionality will be unavailable.

3 Specific Requirements

3.1 Functional Requirements

3.1.1 Student Login with Given Credentials

3.1.1.1 Introduction

Defines how a student logs into the system using credentials provided by the institution. Access is allowed only after successful authentication.

3.1.1.2 Inputs

- Student username or ID
- Password

3.1.1.3 Processing

- Student enters username and password.
- System verifies credentials from stored records.
- If valid, the student is logged in.
- If invalid, an error message is displayed.

3.1.1.4 Outputs

- Successful login confirmation
- Access to student dashboard
- Error message for invalid credentials

3.1.1.5 Performance Requirements

- Login should complete within 2–3 seconds.
- Multiple students can log in simultaneously.

3.1.1.6 Design Constraints

Standards Compliance

System should follow secure authentication standards.

Hardware Limitations

System performance may depend on internet speed and device capability.

3.1.1.7 Attributes

Security

Credentials must be protected from unauthorized access.

Maintainability

Login module should be easy to update without affecting other modules.

3.1.1.8 Other Requirements

Database

Student login details stored securely.

Operations

System administrators manage login permissions.

Site Adaptation

System supports login from different network environments.

3.1.2 Uploading Quiz and Assignment Answer Images

3.1.2.1 Introduction

Students can select a course, open the quiz/assignment upload page, and upload image files of handwritten answers.

3.1.2.2 Inputs

- Student ID
- Course selection
- Quiz or assignment selection
- Image files of answers

3.1.2.3 Processing

- System allows image uploads.
- Verifies file format and deadline.
- Valid files stored securely.
- Invalid or late submissions generate an error.

3.1.2.4 Outputs

- Upload success confirmation
- Stored answer files
- Error messages if upload fails

3.1.2.5 Performance Requirements

- Upload completes within 3–5 seconds depending on file size.
- Supports multiple concurrent uploads.

3.1.2.6 Design Constraints

Standards Compliance

File handling follows storage and security guidelines.

Hardware Limitations

Upload speed may depend on device and internet quality.

3.1.2.7 Attributes

Security

Uploaded files protected from unauthorized access.

Maintainability

Supports future file formats.

3.1.2.8 Other Requirements

Database

Files stored securely.

Operations

System logs upload activity.

Site Adaptation

Supports classroom and remote environments.

3.1.3 Viewing Quiz/Assignment Report and Class Performance Graph

3.1.3.1 Introduction

Students can view their performance report and class performance graph.

3.1.3.2 Inputs

- Student ID
- Selected quiz or assignment
- System-generated evaluation data

3.1.3.3 Processing

- Retrieves student's own performance report.
- Retrieves class performance data.
- Displays summary graph.

3.1.3.4 Outputs

- Student performance report
- Class performance graph
- Error messages if data unavailable

3.1.3.5 Performance Requirements

Reports load in 2–4 seconds; graphs display smoothly.

3.1.3.6 Design Constraints

Standards Compliance

Privacy and data protection standards followed.

Hardware Limitations

Graph rendering depends on device capability.

3.1.3.7 Attributes

Security

Students cannot access others' reports.

Maintainability

Reporting module allows future enhancements.

3.1.3.8 Other Requirements

Database

Reports stored securely.

Operations

Tracks when reports are viewed.

Site Adaptation

Viewable from multiple devices/locations.

3.1.4 Administrator Assign Courses and Create Faculty Credentials

3.1.4.1 Introduction

Administrator assigns courses to faculty and generates credentials.

3.1.4.2 Inputs

- Faculty details (name, email, department)
- Course information (code, name, semester)

3.1.4.3 Processing

- Select “Assign Courses to Faculty”.
- Associate selected courses with faculty.
- Generate secure login credentials.

3.1.4.4 Outputs

- Confirmation of course assignment and credentials
- Error messages if invalid

3.1.4.5 Performance Requirements

Completion within 2–3 seconds per faculty.

3.1.4.6 Design Constraints

Standards Compliance

AES-256 encryption for credentials.

Hardware Limitations

Server supports up to 500 concurrent faculty accounts.

3.1.4.7 Attributes

Security

Only administrators can assign/create credentials.

Maintainability

Easy to update assignment logic.

3.1.4.8 Other Requirements

Records stored securely in database.

3.1.5 Administrator Enroll Students and Create Credentials

3.1.5.1 Introduction

Administrator enrolls students and generates login credentials.

3.1.5.2 Inputs

- Student details (name, email, registration number)
- Course selection

3.1.5.3 Processing

- Enroll students in courses.
- Generate credentials automatically.

3.1.5.4 Outputs

- Enrollment confirmation
- Credential creation confirmation
- Error if duplicate enrollment

3.1.5.5 Performance Requirements

Completion within 2–3 seconds per student.

3.1.5.6 Design Constraints

Standards Compliance

Secure password and data storage.

Hardware Limitations

Support batch enrollment of 500 students.

3.1.5.7 Attributes

Security

Credentials confidential to students.

Maintainability

Bulk enrollment easily updated.

3.1.5.8 Other Requirements

Database stores enrollment/credential records securely.

3.1.6 Administrator Credential Update

3.1.6.1 Introduction

Administrator updates credentials if lost or forgotten.

3.1.6.2 Inputs

- Username or email
- New password or credential details

3.1.6.3 Processing

- Verify user identity
- Update credential in database
- Send confirmation

3.1.6.4 Outputs

- Confirmation of successful update
- Error if verification fails

3.1.6.5 Performance Requirements

Completion under 2 seconds.

3.1.6.6 Design Constraints

Admin-only access, encrypted storage.

3.1.6.7 Attributes

Security

Only authorized admin updates credentials.

Maintainability

Audit logs maintained.

3.1.6.8 Other Requirements

Update records in User Credentials table.

3.1.7 Teacher Login and Dashboard**3.1.7.1 Introduction**

Teachers log in and view assigned courses.

3.1.7.2 Inputs

Teacher username and password.

3.1.7.3 Processing

Authenticate credentials and display assigned courses.

3.1.7.4 Outputs

Teacher dashboard, login confirmation or error.

3.1.7.5 Performance Requirements

Authentication within 1–2 seconds; dashboard loads within 2 seconds per course.

3.1.7.6 Design Constraints

Web browser compatibility; secure HTTPS login.

3.1.7.7 Attributes

Session timeout for security; modular dashboard for maintainability.

3.1.7.8 Other Requirements

Database links teacher table with course assignment.

3.1.8 Teacher Create Assessment

3.1.8.1 Introduction

Teachers create assessments, set portal times, and enable student uploads.

3.1.8.2 Inputs

- Assessment number
- Portal start and end times
- Checkbox for student uploads

3.1.8.3 Processing

- Select course, create assessment
- Popup collects details
- Activate submission portal if enabled

3.1.8.4 Outputs

Confirmation of creation; active submission portal.

3.1.8.5 Performance Requirements

Complete in under 3 seconds; support 200+ simultaneous submissions.

3.1.8.6 Design Constraints

Popup UI follows accessibility guidelines; server handles real-time uploads.

3.1.8.7 Attributes

Only authorized teachers can create assessments; modular for maintainability.

3.1.8.8 Other Requirements

Database stores assessments and portals.

3.1.9 Automated Assessment Checking and Report Generation

3.1.9.1 Introduction

System automatically checks student assessments and generates reports.

3.1.9.2 Inputs

Teacher-uploaded keys; student submissions.

3.1.9.3 Processing

Compare answers with keys; generate student and teacher reports.

3.1.9.4 Outputs

Individual student reports; class performance reports.

3.1.9.5 Performance Requirements

50 submissions checked in 5 seconds; 100 reports in 10 seconds.

3.1.9.6 Design Constraints

Reports in PDF/HTML; server supports bulk processing.

3.1.9.7 Attributes

Accessible only to authorized users; modular marking logic.

3.1.9.8 Other Requirements

Database tables: AnswerKeys, Student_Submissions, Reports.

3.1.10 Teacher Control Over Student Marks Visibility

3.1.10.1 Introduction

Teachers choose whether students can see their marks.

3.1.10.2 Inputs

Teacher selection (show/hide marks).

3.1.10.3 Processing

Record teacher preference; control visibility in dashboard.

3.1.10.4 Outputs

Updated student dashboard; confirmation to teacher.

3.1.10.5 Performance Requirements

Changes reflected within 1–2 seconds.

3.1.10.6 Design Constraints

Only authorized teachers can modify; maintain report integrity.

3.1.10.7 Attributes

Prevent unauthorized access; easily toggle multiple assessments.

3.1.10.8 Other Requirements

Database field indicates visibility status.

3.2 External Interface Requirements

3.2.1 User Interfaces

The system shall provide a user-friendly web-based graphical user interface accessible via standard web browsers. For mobile devices, a responsive interface shall adapt to screen size. The interface shall be intuitive such that users can understand the system within 2–3 uses. Separate dashboards shall be provided for faculty, administrators, and students, showing role-specific functionalities.

3.2.1.1 Performance Requirements

- User interfaces shall load within 2–3 seconds under normal network conditions.
- The system shall handle multiple users simultaneously without performance degradation.
- The system shall be operational 99.9% of the time.
- Teacher and student logins shall authenticate within 2 seconds.
- Assessment submission portals shall activate within 1 second of creation.
- CPU usage shall not exceed 80% under peak load conditions.
- Memory consumption shall be limited to 50 MB per active session.
- The system shall automatically recover from temporary server or network failures within 1 minute without data loss.

3.2.1.2 Design Constraints

Standards Compliance

- The user interface shall comply with IEEE software documentation standards.
- Basic web accessibility standards shall be followed.
- Consistent layout, fonts, and color schemes shall be maintained.
- The system shall adhere to international standards for software quality, including usability, reliability, and maintainability.
- Secure communication protocols shall be used for all data transmission.
- Information security management standards shall be followed.

3.2.2.2 Hardware Limitations

- Hardware shall allow upgrades to CPU, RAM, and storage for increased student load or additional courses.
- Minimum 500 GB storage for student submissions, assessments, and reports.
- Interface shall operate on standard computers, tablets, and smartphones.
- No specialized display or input hardware shall be required.
- Minimum 16 GB RAM to support database operations, AI processing, and multiple sessions.

3.2.1.3 Attributes

Security

- All users shall authenticate using a unique username and password.
- Passwords shall be stored in encrypted form using industry-standard encryption.
- Only administrators may create, update, or reset user credentials.
- Student assessment data, marks, and reports shall be accessible only to authorized users.
- Backups shall be securely stored and protected against unauthorized access.

Maintainability

- System shall maintain error and activity logs for troubleshooting.
- System updates shall be possible with minimal downtime.
- Fault isolation within modules to prevent system-wide failures.
- User interface components shall be modular and reusable.
- UI updates and enhancements shall be easily implementable without affecting core functionality.

3.2.1.4 Other Requirements

Database

- Store course details, faculty-course assignments, and student enrollments.
- Store assessment details, including number, course association, submission timing, and upload permissions.
- Store teacher-uploaded answer keys securely for automated evaluation.
- Support configurable data retention policies for archiving old assessments and reports.

Operations

- Administrators shall manage users and system settings.
- Teachers and students shall perform academic activities via the interface.

Site Adaptation

- Support deployment in different institutional environments, including on-premise and cloud.
- Operate effectively under varying network conditions (LAN, internet), with configurable network settings.
- Allow customization of academic terms, grading schemes, and course structures.
- Support configuration for different languages, time zones, and date formats.
- Allow site-specific configuration of data storage locations and backup policies.

3.2.1.5 Hardware Interfaces

- Operate on standard computing devices: desktops, laptops, tablets, and smart-phones.
- Input devices: keyboard, mouse, and touch input.
- Output devices: standard display screens.
- System shall function on devices with internet connectivity.
- Mobile cameras shall be supported for image uploads.
- System shall detect and handle hardware-related errors.

3.2.1.6 Software Interfaces

- Operate on Linux and Windows server OS without OS-specific modifications.
- Support web browsers: Chrome, Firefox, Edge.
- Backend server for processing business logic.
- Database management system for storing user and academic data.
- AI evaluation module for automated answer assessment.
- All software interfaces shall follow secure API communication standards.

3.2.1.7 Communications Interfaces

- Use HTTP/HTTPS protocols for communication.
- HTTPS shall be used for transmitting sensitive data.
- Support standard internet connections.
- Client-server communication shall be reliable and secure.
- Use TCP/IP as the underlying network protocol.
- Provide standardized error codes and messages for communication failures.

4 Appendix

4.1 Use Case Diagram

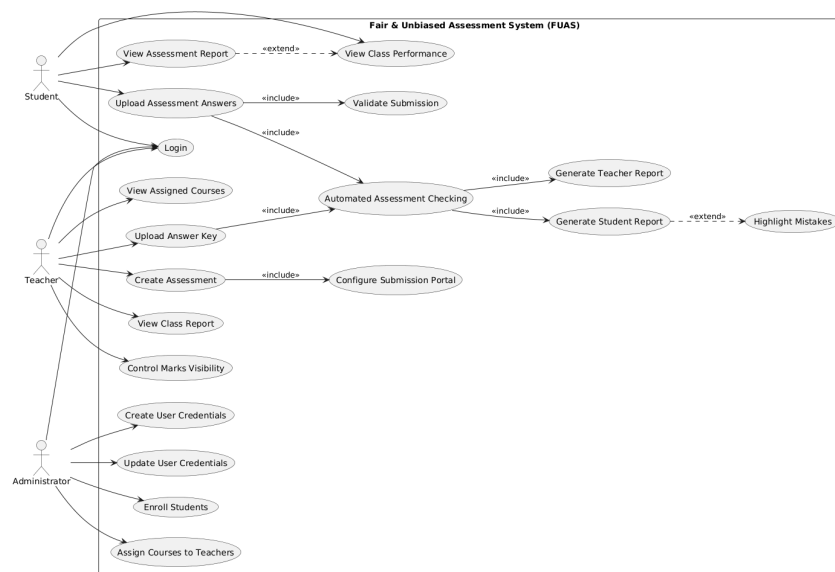


Figure 1: UseCase

4.2 Context Diagram

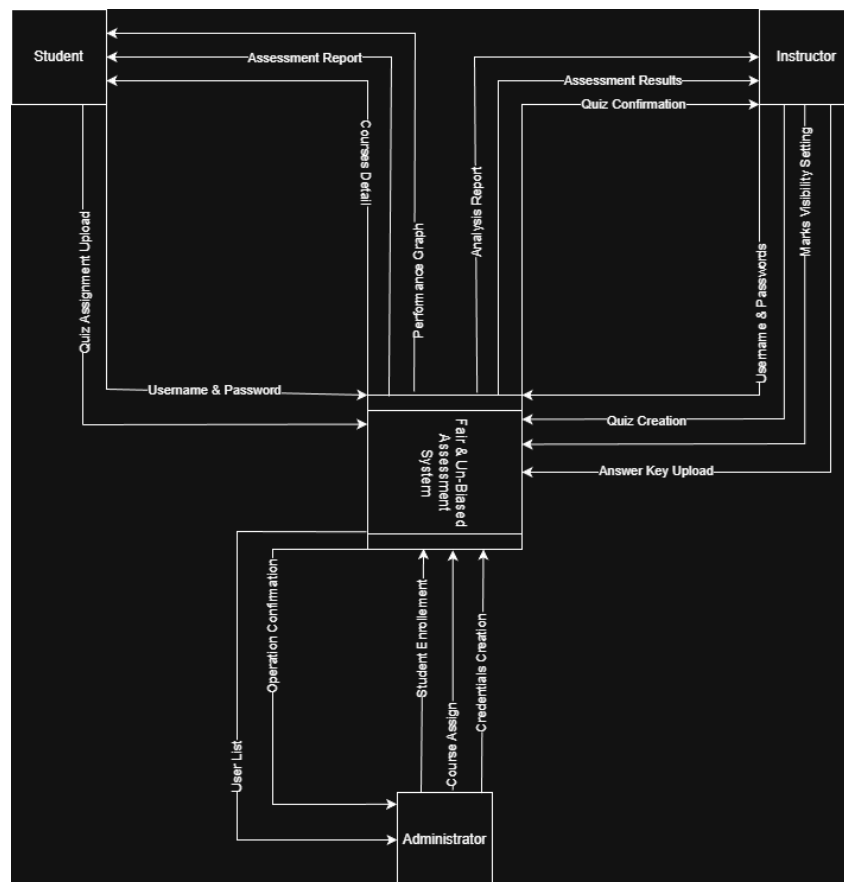


Figure 2: DFD Diagram Level 0