Software Requirements Specification (SRS)

VirtualExam – A Web-Based Examination Management Platform

Course: CSE327 – Software Engineering
North South University, Bangladesh
Semester: Summer 2025

Prepared by Group Members:

Name	ID
Emtiaz Mahamud Emon	2121595642
Israt Jahan Asha	1921758042
Jannatul Ferdous Ruma	2031425642
Sofwat Tahsin	2012561042
Isha Bin Mahmud Mahi	2212373042

Faculty: Dr. Md Musfique Anwar (MMA1)

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) describes the functional and nonfunctional requirements for the **VirtualExam** platform. VirtualExam is a web-based online examination system that facilitates conducting and managing multiple-choice question (MCQ) based exams. It provides interfaces and functionalities tailored for two primary user groups: **students** and **administrators/instructors**.

This document serves as a blueprint for the development, testing, and deployment of the system and will be referenced by the development team, testers, QA engineers, and project managers.

1.2 Intended Audience

This SRS is designed for the following stakeholders:

- **Developers** – to implement the system based on detailed requirements
- **Testers/QA** – to validate the implementation against expected outcomes
- **Project Managers** – to oversee deliverables and timelines
- **Instructors/Admins** – to operate, manage, and evaluate student performance
- **Students** – as end-users who will access exams, view results, and submit feedback
- **University Evaluators** – for reviewing and grading the final capstone project

1.3 Intended Use

The SRS will be used to:

- Guide the development of a full-stack academic quiz system
- Support backend logic and frontend interface for both students and admins
- Enable automation of exam processes and digitization of feedback/results
- Define test cases for evaluation and user acceptance

1.4 Product Scope

VirtualExam simplifies the examination process in educational institutions by offering features like course enrollment, exam creation, automated result calculation, and feedback collection. It replaces traditional paper-based exams with a responsive, secure, and user-friendly interface for students and an admin dashboard for instructors.

The platform focuses on multiple-choice questions (MCQs) and ensures exam integrity through controlled access, real-time tracking, and time limits.

1.5 Risk Definitions

- **Data Loss Risk**: If not properly backed up, student exam data may be lost
- **Security Risk**: Unverified users may access or alter exams without secure login
- **Cheating Risk**: Without question randomization, students may cheat during exams
- **Scalability Risk**: The system might not support large user loads
- **Incomplete Exams**: Poor network can cause mid-exam failures

- **Administrative Overhead**: Manual approval of student requests may cause delays

2. Overall Description

2.1 User Classes and Characteristics

2.1.1 Admin/Instructor

- Can create and manage courses
- Can add students, exams, and announcements
- Can view feedback and student performance
- Requires moderate technical knowledge

2.1.2 Student

- Registers and logs in to access exams
- Takes quizzes and views results
- Provides anonymous feedback
- Requires intuitive, minimal interface

2.2 User Needs

User	Needs
Admin	Add/edit/delete courses, exams, students, results, announcements
Student	Attend exams, track results, submit feedback
Both	Fast, secure access; intuitive design; mobile-friendly layout

2.3 Operating Environment

Component	Details
Frontend	HTML, CSS, JS, Bootstrap
Backend	PHP
Database	MySQL
Server	Apache (XAMPP / LAMP)
Supported OS	Windows, macOS, Linux, Android
Browsers	Chrome, Firefox, Edge, Safari
Devices	Desktop, Laptop, Mobile

2.4 Constraints

- Platform restricted to MCQ-type exams only
- Limited support for concurrent exams unless optimized
- Relies on stable internet connection
- Session timeout for unresponsive users

2.5 Assumptions

- Students will use their own devices to attend exams
- Admins have authority to approve/reject requests
- All users have basic digital literacy
- Exams will follow a fixed time and date structure

3.1 Functional Requirements

(10 key features assigned 2 per group member)

FR1: Student Registration Request

- **Actor:** Student
- **User Story:**
  As a student, I want to register on the platform by submitting my basic information so that I can request access to take exams.
- **Acceptance Criteria:**
  - Student submits name, email, password, and ID.
  - Admin is notified and reviews the request.
  - Student receives approval/rejection via email.

FR2: Student Login and Password Reset

- **Actor:** Student

- **User Story:**
  As a student, I want to securely log in and reset my password if I forget it, so that I can access my dashboard without administrative help.

- **Acceptance Criteria:**

    - Login validated using secure credentials.

    - Forgotten passwords are recovered via email reset link.

    - Account locks temporarily after multiple failed attempts.

FR3: Course Management by Admin

- **Actor:** Admin

- **User Story:**
  As an admin, I want to add, view, or delete courses so that I can organize and assign exams based on course subjects.

- **Acceptance Criteria:**

    - Course creation includes title, code, and description.

    - Admin can edit or remove a course.

    - Course list is visible in the dashboard.

FR4: Student Enrollment Management

- **Actor:** Admin

- **User Story:**
  As an admin, I want to enroll students in specific courses so that only authorized users can take the appropriate exams.

- **Acceptance Criteria:**

    - Admin can add/remove students from courses.

    - Enrolled students are listed under each course.

    - Changes are updated in real-time.

FR5: Exam Creation

- **Actor:** Admin

- **User Story:**
  As an admin, I want to create MCQ-based exams and assign them to courses so that students can participate in scheduled assessments.

- **Acceptance Criteria:**

    - Exams include duration, marks, and time limits.

    - Admin can set exams as active/inactive.

    - Questions are securely saved in the database.

FR6: Exam Participation

- **Actor:** Student
- **User Story:**
  As a student, I want to attend available exams within the given schedule so that I can complete my assessments online.
- **Acceptance Criteria:**
  - Timer initiates upon starting the exam.
  - One question per screen with navigation.
  - Exam auto-submits after timeout or completion.

FR7: View Results

- **Actor:** Student
- **User Story:**
  As a student, I want to view my exam results immediately after submission so that I can assess my performance.
- **Acceptance Criteria:**
  - Results display score, percentage, and question-wise analysis.
  - System ensures result integrity and accuracy.
  - PDF download option is available.

FR8: Result Dashboard by Course

- **Actor:** Admin
- **User Story:**
  As an admin, I want to view all students' results for each course so that I can analyze performance and maintain records.
- **Acceptance Criteria:**
  - Tabular view with filters by course and date.
  - Downloadable in CSV or PDF formats.
  - Includes average and highest marks.

FR9: Submit Feedback

- **Actor:** Student
- **User Story:**
  As a student, I want to submit feedback on the exam experience so that instructors can improve future assessments.
- **Acceptance Criteria:**
  - Feedback includes optional star rating and comments.
  - Anonymous submission is allowed.
  - Feedback is stored for admin review.

FR10: Post Announcements

- **Actor:** Admin
- **User Story:**
  As an admin, I want to post important course announcements so that students

are informed of exams and other updates.

- **Acceptance Criteria:**
    - Announcement includes title, content, and timestamp.
    - Visible on student dashboards.
    - Admin can edit or remove existing announcements.

3.2 Non-Functional Requirements

1. Performance

- Load exam page under 2 seconds
- Must support 100+ concurrent users during peak

2. Security

- Encrypted passwords using SHA-256
- Session expiration after 15 min idle time
- Role-based permission enforcement

3. Usability

- Interface must be responsive and mobile-friendly
- At least 90% student satisfaction from usability surveys

4. Reliability

- 99.5% uptime

- Auto-save answers every 30 seconds during exams

5. Maintainability

- Modular backend code with clear naming conventions
- Source control using Git (GitHub private repo)

6. Scalability

- Must support growth from single course to multiple
- DB design should allow flexible question pool management

4. Appendix

- **Diagrams**: Use Case Diagram, Entity Relationship Diagram (ERD), and Flowcharts are included separately to support system design and logic.
- **Tools Used**: HTML, CSS, JavaScript, PHP, Bootstrap, MySQL, Apache Server, Git, Visual Studio Code.
- **Repository**: The complete project is maintained under a private GitHub repository.
- **Testing**: Manual black-box testing was performed for feature validation.
- **Future Enhancements**: Include camera-based proctoring, coding contests, payment gateway integration, and mobile app support.