

College Examination System

Scenario

Admin, Lecturer, and Student interact with the system to manage exams, perform examinations, and handle accounts and results.

Triggering Event

- Admin wants to manage users.
- Lecturer wants to create or publish an exam.
- Student wants to start an available exam.

Brief Description

The system enables Admin to manage users, Lecturer to create and publish exams, and Student to take exams and view results. All actions are secured by roles and logged for auditing.

Actors

1- Admin 2- Lecturer 3- Student

Related Use Cases

Manage Users, Create Exam, Take Exam, Publish Results, View Grades

Preconditions

- User must be logged in.
- User must have valid permissions.
- Database server must be available.
- Exam must be activated (for Students).

Postconditions

- All actions are saved.
- Logs are recorded.
- Student exam attempts are stored.
- Grades are updated after evaluation.

Flow of Activities

1) Actor: Admin

Admin manages users, subjects, approves grades, and publishes results.

System (for Admin)

- 1.1 Shows Admin Dashboard with: Manage Users, Manage Subjects, Review/Approve Grades, Publish Results, View Reports.
- 1.2 Displays user list, search, Add/Edit/Delete forms.
- 1.3 Validates data, checks duplicates, saves changes, sends activation emails, logs actions.
- 1.4 Displays subject list, allows Add Subject, Assign Lecturer/Students, validates and saves.
- 1.5 Shows pending grades for approval/rejection and logs all decisions.
- 1.6 Publishes results, sends notifications, updates records, and logs action.
- 1.7 Provides reports and audit logs (CSV/PDF export).

2) Actor: Lecturer

Lecturer creates exams, adds questions, sets answers/duration, modifies exams, views grades.

System (for Lecturer)

- 2.1 Shows Lecturer Dashboard with exam tools.
- 2.2 Displays Create Exam form, question bank, add/edit question UI.
- 2.3 Validates exam fields and question completeness.
- 2.4 Saves & publishes exam, notifies students, logs the action.
- 2.5 Tracks exam versions, updates schedule, sends notifications if modified.
- 2.6 Shows student attempts, auto-graded results, manual grading panel.
- 2.7 Provides analytics (average score, question time, item analysis).

3) Actor: Student

Student views available exams, starts exams, answers questions, submits attempts, views results.

System (for Student)

- 3.1 Displays Student Dashboard with exams, attempts, results, feedback options.
- 3.2 Shows available exams based on eligibility.
- 3.3 Verifies rules, starts timer, locks navigation, logs attempt start.
- 3.4 Auto-saves answers, tracks timestamps, prevents multiple attempts.
- 3.5 Finalizes attempt, auto-grades, queues manual grading, shows confirmation.
- 3.6 Publishes results, sends notifications, allows re-correction requests.
- 3.7 Handles connection loss with auto-save and session recovery.

Exception Conditions

General system errors such as invalid inputs, unauthorized access, unavailable resources, network interruptions, or any unexpected issue preventing successful operation.

Use Case Name	College Examination System	
Scenario:	Admin, Lecturer, and Student interact with the system to manage exams, perform examinations, and handle user accounts and results.	
Triggering Event:	• Admin wants to manage users, • Lecturer wants to create or publish an exam, • Student wants to start an available exam.	
Brief Description:	The system allows the Admin to manage users, the Lecturer to create and publish exams, and the Student to take exams and view results. All actions are logged and permissions are controlled by roles.	
Actors:	1-Admin 2-Lecturer 3-Student	
Related Use Cases:	Manage Users Create Exam Take Exam Publish Results View Grades	
Stakeholders	1-Admin 2-Lecturer 3-Student	
Preconditions:	1. User must be logged in, 2. User must have valid permissions based on role, 3. Database must be accessible, 4. Exam must be published and active (for students).	
Preconditions:	1. All actions such as creating users or exams are saved, 2. Activity logs are recorded, 3. Student exam attempts are stored, 4. Student results are updated after grading.	
Flow of Activities:	Actor	System
Exception Conditions	1. Admin / Lecturer / Student accesses the system.	1.1 System displays the dashboard based on user role.
	2. Admin manages the system resources (users, subjects, grades).	2.1 Student "Admin Dashboard" with quick links: Manage Users, Manage Subjects, View Exam Grades, Manage Grades, Publish Results, View Reports. 2.2 When Admin opens Manage Users → shows user list, search filter, Add/Update/Delete forms. 2.3 Validation logic: checks duplicates, name changes, sends activation email for new users. Logs every action. 2.4 When Admin opens Manage Subjects → shows subject list, show Add/Update/Delete buttons, Assign Subjects, validate and save. 2.5 When Admin opens Review/Approve Grades → shows pending grades, provides Approve/Reject controls, updates bulk actions, logs decisions. 2.6 When Admin starts Publish Results → system writes approved grades, updates student records, triggers email notifications and audit logs. 2.7 Provides reports and audit logs (exportable CSV/JSON) for all admin actions.
	3. Lecturer creates and manages exams: adds questions, sets answers/duration, modifies exams, and views student grades.	3.1 Displays "Lecturer Dashboard" with: Create Exam, Modify Exam, View Student Grades, View Reports, Add Questions. 3.2 When Lecturer opens Create Exam → shows exam form (title, subject, duration, availability window, shuffle settings, question bank access, auto-kill question UI). 3.3 Validates exam fields and question completeness, warns on missing required fields. 3.4 On Publish → system saves exam, sets visibility rules, sends email notifications, updates statistics (if enabled). Logs the publish action. 3.5 When Lecturer modifies exam (before publish or if allowed after) → system tracks version, validates changes, updates scheduled availability and re-activates student access if necessary. 3.6 When Lecturer requests View Student Grades → system displays student list and grades, sorts, logs requested access for manual grading, allows archiving, manual marks, and logs grading actions. 3.7 Provides exam analytics: time per question, average score, item analysis, overall report options.
	4. Student browses available exams, starts a scheduled exam, answers questions (auto-save), submits attempt, and views results/feedback.	4.1 Student "Student Dashboard" with Available Exams, In-Progress Attempts, Results, Request Reconnection, Send Feedback. 4.2 When Student opens Available Exams → shows exam list filtered by enrollment/eligibility, shows instructions and start button. 4.3 On Start Exam → system verifies eligibility (enrollment rules, time window), starts secure timer, locks navigation (prevent), and logs start time. 4.4 While taking exam → system auto-saves answers (auto-save), tracks time/attempt per question, prevents multiple concurrent attempts (per policy), and shows remaining time. 4.5 On Submit → system finalizes attempt, triggers auto-saving for report items, causes scheduled items for future review, marks submission time, and shows confirmation. 4.6 After grading → system makes marks available (according to publish rules), notifies student, allows student to request reconnection or send feedback, logs requests and routes them to Lecturer/Admin. 4.7 If connection issues occur → system uses local connectivity and secure storage where possible, logs disconnection and restores answers on reconnection.
	Invalid or missing data, unauthorized access attempts, system unavailability, connectivity issues, or any failure preventing the completion of the requested action.	

