Paperless Examination System (Objective + Subjective)

1. Overview

This document defines the **complete end-to-end design and workflows** of a **paperless examination system** supporting **objective, subjective, and mixed-mode exams** across **multiple universities**.

It emphasizes **who does what, when, and how data flows** between roles, with strict security, RBAC, AI-assisted evaluation, and auditability.

1A. Master Workflow Map (At a Glance)

Actors: Super Admin → University Admin → Exam Dept → Dean → HOD → Teacher → Student → Evaluator → Exam Dept

High-Level Flow:

1. Platform & University Setup
2. Exam Configuration
3. Paper Creation & Approval
4. Student Allocation & Exam Conduction
5. Evaluation & AI Assistance
6. Result Publishing & Scrutiny

---

2. Stakeholders & Roles

2.1 Platform-Level Roles

Super Admin

- Creates & manages universities
- Defines global roles and permissions
- Audits compliance and security

2.2 University-Level Roles

University Admin / Chief Dean

- Manages exam system for a university
- Creates schools
- Assigns School Deans

Exam Department (Controller of Examination)

- Owns the complete exam lifecycle
- Opens exam windows
- Controls paper flow, scheduling, result publishing & scrutiny

---

2.3 Academic Roles

School Dean

- Selects courses & paper types
- Final physical verification of papers

Deputy Dean

- Assists School Dean
- Manages coordination & approvals

HOD (Head of Department)

- Assigns paper setters & evaluators
- Verifies questions & evaluations

Teacher / Faculty

- Paper setter
- Answer evaluator

2.4 Student

- Appears in exam
- Views schedule, room, result & scrutiny

---

3. Role-Based Access Control (RBAC) – Permission-Centric Model

This system follows a **pure permission-driven architecture**. **Roles are not hard-coded** (Dean, HOD, Teacher are just labels). Actual power comes from **assigned permissions**.

Roles like **Deputy Dean, HoS (Head of Section), Senior Faculty, External Evaluator** can be created dynamically and mapped to permissions.

3.1 Permission Categories

Permissions are grouped into logical sets for the complete examination lifecycle.

A. Academic Structure Permissions

Permission Code	Description
CREATE_SCHOOL	Create schools
CREATE_DEPARTMENT	Create departments
CREATE_PROGRAMME	Create programmes
CREATE_COURSE	Create courses
ASSIGN_ACADEMIC_ROLE	Assign Dean / HOD / Teacher

B. Exam Configuration Permissions

Permission Code	Description
CREATE_EXAM	Create exam event
OPEN_EXAM_WINDOW	Open exam configuration window
SELECT_PAPER_TYPE	Select subjective/objective/mix
SET_EXAM_DURATION	Define exam duration
FREEZE_EXAM_CONFIG	Lock exam settings

C. Paper Structure & Blueprint Permissions

Permission Code	Description
SET_PAPER_STRUCTURE	Define marks distribution, sections
SET_NO_OF_SETS	Define number of paper sets
DEFINE_BLOOMS_DISTRIBUTION	Set Bloom’s taxonomy mapping
DEFINE_DIFFICULTY_RATIO	Set difficulty level ratio

D. Paper Setter Management Permissions

Permission Code	Description
ASSIGN_PAPER_SETTER	Select users for paper setting
VIEW_PAPER_SETTERS	View assigned setters

Permission Code	Description
REMOVE_PAPER_SETTER	Remove/replace setter

E. Question Authoring & Moderation Permissions

Permission Code	Description
CREATE_QUESTION	Add questions
EDIT_QUESTION	Modify questions
SUBMIT_QUESTION	Submit questions
VERIFY_QUESTION	Approve / reject questions
MANAGE_QUESTION_BANK	Push to question bank

F. Paper Generation & Approval Permissions

Permission Code	Description
GENERATE_PAPER	Auto-generate papers
VIEW_GENERATED_PAPER	View paper
APPROVE_PAPER	Final approval
SUBMIT_FINAL_PAPER	Lock & submit paper

G. Exam Conduction Permissions

Permission Code	Description
UPLOAD_STUDENT_LIST	Upload student Excel
ASSIGN_ROOM	Assign room numbers
OPEN_EXAM	Open exam window for students
CLOSE_EXAM	Close exam

H. Evaluation & AI Permissions

Permission Code	Description
ASSIGN_EVALUATOR	Assign evaluator
DEFINE_CHECKING_PATTERN	Set evaluation pattern
VIEW_ANSWER_SCRIPT	View scripts (anonymous)
EVALUATE_SCRIPT	Evaluate answers
ACCEPT_AI_MARKS	Accept AI-suggested marks
OVERRIDE_AI_MARKS	Manually modify marks

I. Result & Scrutiny Permissions

Permission Code	Description
PUBLISH_RESULT	Publish result
OPEN_SCRUTINY	Open scrutiny window
VIEW_SCRUTINY_SCRIPT	Show answers to student
APPROVE_MARK_CHANGE	Approve mark improvement

J. Audit & Security Permissions

Permission Code	Description
VIEW_AUDIT_LOG	View audit logs
EXPORT_REPORT	Export reports
FORCE_PAPER_LOCK	Emergency paper lock

3.2 Permission-to-Role Distribution (Example)

These are **examples only**. Actual roles are configurable.

Role	Key Permissions
Dean	APPROVE_PAPER, SUBMIT_FINAL_PAPER
Deputy Dean	VIEW_GENERATED_PAPER, DEFINE_BLOOMS_DISTRIBUTION
HOD	SET_PAPER_STRUCTURE, ASSIGN_PAPER_SETTER, VERIFY_QUESTION, ASSIGN_EVALUATOR
Teacher (Setter)	CREATE_QUESTION, SUBMIT_QUESTION
Teacher (Evaluator)	VIEW_ANSWER_SCRIPT, EVALUATE_SCRIPT

3.3 Permission Resolution Logic

1. User logs in
2. System fetches assigned permissions
3. APIs validate permission before execution
4. UI components rendered conditionally

3.4 Key Design Principle

A role without permission has zero power.
A user with permission can perform action regardless of role name.

This ensures:

- Future-proof system
- No hard dependency on academic titles
- Easy onboarding of new roles

---

4. Academic Structure Hierarchy

University → School → Department → Programme → Course → Section

Each level:

- Has metadata
- Has assigned roles
- Is permission-scoped

---

5. Examination Lifecycle – Ultra-Detailed Workflow (Process + Data Flow)

This section explains **every micro-step**, including **state changes, validations, locks, notifications, and data movement**. This is suitable for **actual system implementation**.

Workflow 1: Platform Initialization & Governance

Actor: Super Admin

Steps:

1. Create University (university_id generated)
2. Configure global policies:
    - Exam rules
    - AI usage policy
    - Data retention
3. Create base roles (Dean, HOD, Teacher, Exam Dept, Student)
4. Define permission matrix (checkbox-based)
5. Lock global configuration

System State:

- Platform ready
- University isolated (multi-tenant)

Workflow 2: University Academic Structure Setup

Actors: University Admin / Chief Dean

Steps:

1. Create Schools (school_id)
2. Assign School Deans
3. Create Exam Department users
4. Create Departments under Schools
5. Create Programmes under Departments
6. Create Courses under Programmes
7. Create Sections (optional)
8. Assign:
    - Deputy Deans
    - HODs
    - Teachers

Validation Rules:

- No exam can be created without a complete hierarchy

Workflow 3: Permission-Based Dashboard Rendering

Actors: All roles

Logic:

- User logs in
- Backend resolves permissions
- Frontend dynamically renders:
    - Visible modules
    - Action buttons
    - API access

Outcome:

- Same role may see different dashboards

Workflow 4: Exam Creation & Configuration

Actor: Exam Department

Steps:

1. Create Exam Event
2. Select:
    - Semester / Year
    - Exam category
    - Exam type (Objective / Subjective / Mixed)
    - Default duration
3. Open exam configuration window

State:

- Exam = DRAFT

Workflow 5: Course Selection & Paper Blueprint

Actor: School Dean

Steps:

1. View available courses
2. View courses for exam
3. For each course:
    - Select paper type
    - Define number of sets
    - Define total marks
4. Submit blueprint

State Change:

- Exam = COURSE_LOCKED

Workflow 6: Paper Setter Assignment

Actors: Exam Dept → HOD

Steps:

Steps:

1. View available courses
2. Select courses for exam
3. For each course:
    - Select paper type
    - Define number of sets

    - Define total marks
4. Submit blueprint

State Change:

5. Exam = COURSE_LOCKED
6. Exam Dept opens assignment window
7. HOD views assigned courses
8. HOD assigns:
    - Paper setters
    - Backup paper setters (optional)
9. Submission

Validations:

- Teacher must belong to course

Workflow 7: Question Authoring (Teacher Level)

Actor: Teacher

Steps:

1. Access paper-setting module
2. For each question:
    - Enter question text
    - Assign marks
    - Bloom's taxonomy level
    - Difficulty level
    - Question type
    - Model answer (mandatory for AI)
3. Save as draft OR submit

State:

- Question = SUBMITTED

Workflow 8: Question Moderation & Bank Formation

Actor: HOD

Steps:

1. Review each question
2. Approve / Reject / Modify
3. Approved questions pushed to Question Bank

4. Submit verified pool to Exam Dept

State:

- Question Bank = SEALED

Workflow 9: Automated Paper Generation

Actor: System (Exam Dept Trigger)

Steps:

1. Fetch question bank
2. Apply shuffle algorithm
3. Generate required sets
4. Validate:
    - Bloom distribution
    - Difficulty balance
5. Encrypt papers

State:

- Paper = GENERATED

Workflow 10: Final Paper Approval (High Security)

Actor: School Dean

Steps:

1. Physical presence login (2FA)
2. Decrypt paper
3. Verify
4. Approve

State:

- Paper = LOCKED

Workflow 11: Student & Seating Allocation

Actor: Exam Department

Steps:

1. Upload Excel (students)
2. Validate roll numbers
3. Assign rooms
4. Generate admit mapping

State:

- Exam = READY

Workflow 12: Exam Conduction (Paperless)

Actor: Student

Steps:

1. Login within time window
2. System verifies:
    - Device
    - Identity
3. Display full question paper
4. For each answer:
    - Select question
    - Write via handwriting canvas
    - Add pages dynamically
5. Auto-save + final submit

State:

- Script = SUBMITTED

Workflow 13: Evaluation Assignment

Actors: Exam Dept → HOD

Steps:

1. Assign HOD
2. HOD assigns evaluators

Workflow 14: AI-Assisted Evaluation Pipeline

System AI Flow:

1. Handwritten OCR (AWS Textract)
2. Language detection
3. Translation (AWS Translate)
4. Answer understanding (LLM)
5. Rubric-based scoring

Exam Department : Define checking pattern

Teacher:
6. Accept / override marks

Workflow 15: Result Processing & Scrutiny

Actor: Exam Dept

Steps:

1. Open scrutiny window
2. Reveal Marks to the student
3. Student approves / Student requests recheck
4. Submitted to Exam → HOD → teacher marks changed
5. Update marks
6. Final result Announced

Final State:

- Exam = CLOSED

---

6. Data Integrity, Logs & Compliance

- Every action logged
- Versioned answer scripts
- Immutable audit trail
- AI decision trace stored

---

7. This Documentation Can Be Used As

- Master SRS
- Backend API blueprint
- Patent technical description
- University tender document