# 4. System Features

## 4.1. Cohort Creation and Student Invitation

### 4.1.1 Description and Priority

Enables faculty to create cohorts for assessments and invite students through email links. Priority: High.

### 4.1.2 Stimulus/Response Sequences

- Stimulus: Faculty selects "Create New Cohort" and provides cohort details.

- Response: System generates a unique invitation link for the cohort.

### 4.1.3 Functional Requirements

- REQ-1: The system should allow faculty to create cohorts by providing cohort details (e.g., name, course).

- REQ-2: Upon cohort creation, the system should generate a unique invitation link.

- REQ-3: Faculty should have the option to send invitation emails to students using the generated link.

- REQ-4: Students clicking the invitation link should be directed to join the respective cohort.

## 4.2. Exam Question Types

### 4.2.1 Description and Priority

Enables the creation of various question types, such as multiple-choice questions, text answers, and speech answers. Priority: Medium.

### 4.2.2 Stimulus/Response Sequences

- Stimulus: Faculty selects "Create New Assessment" and chooses question type.

- Response: System displays appropriate input fields based on the selected question type.

### 4.2.3 Functional Requirements

- REQ-5: The system should allow faculty to choose the question type during assessment creation.

- REQ-6: For multiple-choice questions, the system should provide options with the ability to mark the correct answer.

- REQ-7: For text answers, the system should display a text input field for students.

- REQ-8: For speech answers, the system should provide a microphone icon for students to record their response.

- REQ-9: The system should validate and store the selected answer type for each question.

## 4.3. Time Limitations for Question Paper Availability

### 4.3.1 Description and Priority

Limits the availability of the question paper to a specific time window. Priority: High.

### 4.3.2 Stimulus/Response Sequences

- Stimulus: Faculty sets the start and end times for question paper availability.

- Response: System calculates the availability window based on the specified times.

### 4.3.3 Functional Requirements

- REQ-10: The system should allow faculty to set the start and end times for question paper availability.

- REQ-11: When a student accesses the question paper, the system should start a countdown timer.

- REQ-12: The countdown timer should be synchronized with the availability window.

- REQ-13: If the student accesses the question paper close to the end time, the timer should reflect the remaining time.

## 4.4. Answer Evaluation and Automatic Marking

### 4.4.1 Description and Priority

Automatically evaluates students' answers and assigns marks based on faculty-provided responses. Priority: High.

### 4.4.2 Stimulus/Response Sequences

- Stimulus: Student submits an answer for evaluation.

- Response: System compares the student's answer with the correct answer and assigns marks.

### 4.4.3 Functional Requirements

- REQ-14: The system should compare student answers with corresponding faculty-provided answers.

- REQ-15: For each question type, the system should apply appropriate evaluation criteria (e.g., match, word similarity, speech recognition accuracy).

- REQ-16: Marks should be assigned automatically and accurately based on the comparison results.

## 4.5. Student Dashboard and Marks Publication

### 4.5.1 Description and Priority

Enables students to view their marks and provides a complaint dashboard. Priority: Medium.

### 4.5.2 Stimulus/Response Sequences

- Stimulus: Student accesses the dashboard after marks publication.

- Response: System displays the student's marks and options for raising complaints.

### 4.5.3 Functional Requirements

- REQ-17: The system should display students' marks after faculty publication.

- REQ-18: The dashboard should allow students to submit complaints about mark discrepancies.

- REQ-19: After a complaint is submitted, the system should notify the faculty for review.

## 4.6. Continuous Student Verification

### 4.6.1 Description and Priority

Verifies student identity using 3D images captured during sign-up. Priority: High.

### 4.6.2 Stimulus/Response Sequences

- Stimulus: Student starts an exam.

- Response: System activates the camera and microphone for image verification.

### 4.6.3 Functional Requirements

- REQ-20: The system should activate the camera and microphone during exams.

- REQ-21: The system should capture 3D images of the student during sign-up for identity verification.

- REQ-22: During exams, the system should match the live image with the stored 3D image to verify student identity.

# 

# 

# 

# 

# 

# 5. Other Nonfunctional Requirements

## 5.1. Performance Requirements

### - Response Time: The system should respond to user actions, such as submitting answers and navigating between questions, within 2 seconds on average under normal load conditions.

### - Exam Loading Time: The system should load the exam paper within 5 seconds after the student accesses it.

### - Scalability: The system should handle up to 1000 simultaneous users taking exams without significant degradation in performance.

### - Resource Utilization: The system should utilize no more than 80% of available server resources (CPU, memory, bandwidth) during peak usage.

## 5.2. Safety Requirements

### - Data Integrity: The system should ensure that student answers and marks are stored accurately and are not subject to unauthorized tampering or alteration.

### - Identity Verification: The system's image verification process must not compromise student privacy or data security.

### - Error Handling: The system should gracefully handle errors and invalid inputs to prevent crashes or data loss.

### - Backup and Recovery: The system should regularly back up data to prevent data loss in case of system failure.

## 5.3. Security Requirements

### - User Authentication: The system must support secure user authentication and authorization mechanisms, including strong password policies and password hashing.

### - Data Encryption: Student data, including answers and personal information, should be encrypted during transmission and storage.

### - Access Control: Faculty should have access only to their own assessments and cohort data, and students should only access their own exam-related information.

### - Audit Trail: The system should maintain an audit trail of important actions, such as login attempts, data modifications, and assessment creation.

## 5.4. Software Quality Attributes

### - Usability: The system's user interface should be intuitive and easy to navigate, with a learnability score of at least 85% for new users.

### - Reliability: The system should have an uptime of at least 99.5% over any given month, ensuring availability during scheduled maintenance.

### - Maintainability: Code should be well-documented and modular to facilitate future maintenance and updates.

### - Portability: The system should be accessible on major web browsers (Chrome, Firefox, Safari, Edge) and responsive to different screen sizes.

## 5.5. Business Rules

-Faculty members are responsible for assessments and cohorts.

-Students can only access exams for specific cohorts.

-Faculty can adjust marks within a designated timeframe after reviewing complaints.

-Students should not share exam questions during the exam window.

# 6. Other Requirements

## 6.1. Database Requirements

- The system shall utilize a relational database for storing user profiles, assessment details, and student responses.

- The database schema shall be designed to efficiently store and retrieve data related to assessments, cohorts, and student marks.

## 6.2. Internationalization Requirements

- The user interface should be designed to support multiple languages to accommodate international users.

- Textual content, labels, and messages should be designed to be easily translatable.

## 6.3. Reuse Objectives

- Code modules should be designed for reusability to expedite the development of future features or projects.

- Reusable components should be well-documented for ease of integration into other projects.