USE-CASE DISCRIPTION

Use Case Name: Student Registration

ID: UC-SR

Primary Actor: Lecturer

Use Case Type: Detail, Essential

Stakeholders and Interests:

Lecturer: Wants to facilitate student registration efficiently.

System Administrator: Manages the overall CBT application.

Brief Description:

This use case describes the process by which lecturers manage student registration within the Computer-Based Testing (CBT) application. Lecturers take responsibility for registering students, collecting and validating essential information.

Trigger: Lecturer wants to register new students for a test.

Importance Level: High

Type: Essential

Relationships:

Association: Lecturer associates with the CBT system to manage student registrations.

Include:

Individual Student Registration:

Lecturer supplies necessary information for individual student registration.

Matriculation number undergoes validation.

Random password generated and sent via email upon successful validation.

Confirmation message displayed for successful record creation.

Error message displayed in case of validation failure or other errors.

Bulk Student Registration:

Lecturer uploads student records in a CSV file.

Matriculation numbers are verified.

Random passwords generated and sent via email for valid entries.

Confirmation message displayed for successful record creation.

System refrains from generating passwords or sending emails for invalid entries.

Extend: None

Generalization: None

Normal Flow of Events:

Lecturer initiates student registration.

Lecturer provides necessary information for individual student registration.

Matriculation number undergoes validation.

Random password is generated and sent via email upon successful validation.

Confirmation message is displayed for successful record creation.

Alternate Flow : Validation Failure in Individual Registration

Lecturer initiates student registration.

Lecturer provides necessary information for individual student registration.

Matriculation number fails validation.

Error message is displayed, indicating the inability to create the student record.

Normal Flow of Events (Bulk Student Registration):

Lecturer initiates bulk student registration.

Lecturer uploads student records in a CSV file.

Matriculation numbers are verified.

Random passwords are generated and sent via email for valid entries.

Confirmation message is displayed for successful record creation.

Use Case Name: Take Test

ID: UC-TT

Primary Actor: Student

Use Case Type: Detail, Essential

Stakeholders and Interests:

- Student: Wants to take a test using the Computer-Based Testing (CBT) application.

- Lecturer: Requires an efficient and reliable system for test administration and grading.

- System Administrator: Manages the overall functionality and security of the CBT application.

Brief Description:

This use case outlines the process by which a student takes a test within the Computer-Based Testing (CBT) application. The student logs in, initiates the test, answers questions, and submits the test for evaluation.

Trigger: Student wants to start and complete a test.

Importance Level: High

Type: Essential

Relationships:

- Association: Student associates with the CBT system to take a test.

Include:

- Log In:

- Student logs into the CBT application using valid login credentials.

- System authenticates the student's information and provides access to the dashboard.

- Start Test:

- Student initiates a test by clicking the "Start" button.

- Countdown timer begins based on the duration set by the lecturer.

- Answer Questions:

- System presents questions based on the lecturer's settings.

- Navigation through questions is allowed using "Next" or "Previous" buttons.

- Student answers questions within the specified time.

- Submit Test:

- Student clicks the "Submit" button or the test automatically submits when the timer reaches zero.

- System marks the script and calculates the score.

- The score is displayed to the student and sent to their email.

- View Script:

- A pop-up menu appears post-test, prompting the student to view their script.

- Clicking a "View Script" button generates the script.

- Print Script:

- Users can opt to print the script by clicking the "Print" button.

- An error message is displayed in case of printing issues.

Extend: None

Generalization: None

Normal Flow of Events:

1. Student logs into the CBT application.

2. System authenticates student's information and provides access to the dashboard.

3. Student initiates the test by clicking the "Start" button.

4. Countdown timer begins based on the duration set by the lecturer.

5. System presents questions based on lecturer's settings.

6. Navigation through questions is allowed using "Next" or "Previous" buttons.

7. Student answers questions within the specified time.

8. Student submits the test by clicking the "Submit" button or automatically when the timer reaches zero.

9. System marks the script and calculates the score.

10. Score is displayed to the student and sent to their email.

#### Alternate Flow 1: Test Already Taken

1. Student attempts to retake a test.

2. System detects that the test has already been taken.

3. Message is displayed indicating that the test has already been taken.

4. Pop-up menu gives the option to view the script.

#### Alternate Flow 2: Printing Script Error

1. Student attempts to print the script.

2. Error occurs during the printing process.

3. System displays a message indicating "Could Not Print Script."

Use Case Name: Submit Test

ID: UC-ST

Primary Actor: Student

Use Case Type: Detail, Essential

Stakeholders and Interests:

- Student: Wants to submit a completed test within the Computer-Based Testing (CBT) application.

- Lecturer: Requires an efficient and reliable system for collecting and grading student assessments.

- System Administrator: Manages the overall functionality and security of the CBT application.

Brief Description:

This use case outlines the process by which a student submits a completed test within the Computer-Based Testing (CBT) application. The student completes the test, initiates the submission process, and receives feedback on the test results.

Trigger: Student completes answering questions and wants to submit the test.

Importance Level: High

Type: Essential

Relationships:

- Association: Student associates with the CBT system to submit a test.

Include:

- Answer Questions:

- Student answers questions during the test-taking process.

- Initiate Submission:

- Student clicks the "Submit" button or the test automatically submits when the timer reaches zero.

- Evaluate Test:

- System marks the script and calculates the student's score.

- Display Score:

- The score is displayed to the student on the test submission confirmation page.

- Send Score to Email:

- The system sends the test score to the student's email for their records.

- View Script:

- A pop-up menu appears post-test, prompting the student to view their script.

- Clicking a "View Script" button generates the script.

- Print Script:

- Users can opt to print the script by clicking the "Print" button.

- An error message is displayed in case of printing issues.

Extend: None

Generalization: None

Normal Flow of Events:

1. Student completes answering questions during the test.

2. Student initiates the submission process by clicking the "Submit" button or automatically when the timer reaches zero.

3. System marks the script and calculates the score.

4. The score is displayed to the student on the test submission confirmation page.

5. System sends the test score to the student's email for their records.

#### Alternate Flow 1: Error in Sending Email

1. System encounters an error while sending the test score to the student's email.

2. Error message is displayed, indicating the failure to send the test score email.

####

Use Case Name: View Test

ID: UC-VT

Primary Actor: Student

Use Case Type: Detail, Essential

Stakeholders and Interests:

- Student: Wants to review and analyze test results using the Computer-Based Testing (CBT) application.

- Lecturer: Requires students to have access to their test results for self-assessment.

- System Administrator: Manages the overall functionality and security of the CBT application.

Brief Description:

This use case details the process by which a student views their test results within the Computer-Based Testing (CBT) application. After completing a test, the student can access and review their performance.

Trigger: Student wants to view the results of a completed test.

Importance Level: High

Type: Essential

Relationships:

- Association: Student associates with the CBT system to view test results.

Include:

- Log In:

- Student logs into the CBT application using valid login credentials.

- System authenticates the student's information and provides access to the dashboard.

- Access Test Results:

- Student clicks on the "View" button dedicated to accessing test results.

- The system displays the scores of all completed tests.

- Save Scores to CSV:

- Student has the capability to save the scores of all completed tests into a CSV file.

- A dedicated tab provides the option to save the scores.

Extend: None

Generalization: None

Normal Flow of Events:

1. Student logs into the CBT application.

2. System authenticates student's information and provides access to the dashboard.

3. Student clicks on the "View" button dedicated to accessing test results.

4. System displays the scores of all completed tests.

#### Alternate Flow 1: Saving Scores to CSV

1. Student has the option to save the scores of all completed tests into a CSV file.

2. Clicking on the "Save to CSV" button initiates the process

Use Case Name: Print Test

ID: UC-PT

Primary Actor: Student

Use Case Type: Detail, Essential

Stakeholders and Interests:

- Student: Wants to print the test script for a review.

- Lecturer: Requires an efficient and user-friendly system for test administration.

- System Administrator: Manages the overall functionality and security of the Computer-Based Testing (CBT) application.

Brief Description:

This use case outlines the process by which a student prints the test script after completing a test within the Computer-Based Testing (CBT) application. The student reviews their answers and the marked script by generating a printable version.

Trigger: Student wants to print the test script for review.

Importance Level: Medium

Type: Essential

Relationships:

- Association: Student associates with the CBT system to print the test script.

Include:

- Take Test:

- The ability to print the test script is an optional step post-test completion.

- Students take a test, and after the test is completed and marked, they have the option to print the script.

Extend:

- View Script:

- To print the script, the student must first view the script.

- Clicking on the "View Script" button allows the student to review their answers before deciding to print.

Generalization: None

Normal Flow of Events:

1. Student completes a test.

2. After test completion, student has the option to print the script.

3. Clicking the "Print" button generates a printable version of the script.

#### Alternate Flow 1: Viewing Script Before Printing

1. To print the script, the student must first view the script.

2. Clicking on the "View Script" button allows the student to review their answers before deciding to print.

Use Case Name: User Login

ID: UC-UL

Primary Actor: User (Lecturer or Student)

Use Case Type: Detail, Essential

Stakeholders and Interests:

- User: Wants to access the Computer-Based Testing (CBT) application securely.

- System Administrator: Manages user authentication and ensures the security of the CBT application.

Brief Description:

This use case details the process by which a user (either a lecturer or a student) logs into the Computer-Based Testing (CBT) application. The user provides valid login credentials to access the system.

Trigger: User wants to log into the CBT application.

Importance Level: High

Type: Essential

Relationships:

- Association: User associates with the CBT system to log in.

Include:

- Input Login Details:

- User inputs their login credentials, including email and password.

- Authenticate User:

- The system authenticates the provided login information.

- If valid, the user is granted access to the dedicated page based on their user type (lecturer or student).

- If invalid, an error message is displayed, indicating "User login details invalid."

Extend: None

Generalization: None

Use Case Name: Set Questions

ID: UC-SQ

Primary Actor: Lecturer

Use Case Type: Detail, Essential

Stakeholders and Interests:

- Lecturer: Wants to set questions for a test within the Computer-Based Testing (CBT) application.

- System Administrator: Manages the overall functionality and security of the CBT application.

Brief Description:

This use case describes the process by which a lecturer sets questions for a test within the Computer-Based Testing (CBT) application. The lecturer specifies the number of questions and may upload questions in a CSV file.

Trigger: Lecturer wants to prepare questions for a test.

Importance Level: High

Type: Essential

Relationships:

- Association: Lecturer associates with the CBT system to set questions.

Include:

- Specify Number of Questions:

- Lecturer accesses the dedicated tab to specify the desired number of questions for the test.

- The system validates the entered number, displaying a success message for a valid entry and an error message for an invalid one.

- CSV File Upload:

- Lecturer can upload questions in a CSV file.

- The application verifies the absence of duplicate questions in the uploaded file.

- If successful, a confirmation message is displayed, indicating the creation of the question bank.

- In case of duplicates or other issues, an error message is presented.

Extend: None

Generalization: None

Use Case Name: User Authentication

ID: UC-UA

Primary Actor: User (Lecturer or Student)

Use Case Type: Detail, Essential

Stakeholders and Interests:

- User: Wants secure access to the Computer-Based Testing (CBT) application.

- System Administrator: Manages the overall security and access control of the CBT application.

Brief Description:

This use case outlines the process by which a user, either a lecturer or a student, authenticates themselves to access the Computer-Based Testing (CBT) application. Authentication ensures that only authorized users can log in and interact with the system.

Trigger: User attempts to log into the CBT application.

Importance Level: High

Type: Essential

Relationships:

- Association: User associates with the CBT system to authenticate and access the application.

Include:

- Input Login Details:

- User inputs their login credentials, including email and password.

- Authenticate Information:

- System verifies the provided login details against stored user data.

- Access is granted for valid credentials.

- Access Dashboard:

- Upon successful authentication, the system provides access to a dedicated dashboard based on the user type (lecturer or student).

- Error Handling:

- In case of invalid login details, the system displays an error message indicating "User login details invalid."

Extend: None

Generalization: None

Normal Flow of Events:

User inputs their login credentials, including email and password.

System verifies the provided login details against stored user data.

Access Dashboard:

Upon successful authentication, the system provides access to a dedicated dashboard based on the user type (lecturer or student).

Alternate Flow: Invalid Login Credentials:

If the entered login credentials are invalid:

The system displays an error message indicating "User login details invalid."

Use Case Name: Upload CSV Questions

ID: UC-UCQ

Primary Actor: Lecturer

Use Case Type: Detail, Essential

Stakeholders and Interests:

- Lecturer: Aims to efficiently upload a set of questions in bulk for a test.

- System Administrator: Manages the overall functionality and security of the CBT application.

Brief Description:

This use case describes the process by which a lecturer uploads a set of questions in CSV (Comma-Separated Values) format to the Computer-Based Testing (CBT) application. The system verifies the absence of duplicate questions, uploads the questions, and confirms the creation of the question bank.

Trigger: Lecturer wants to add questions to the question bank by uploading a CSV file.

Importance Level: High

Type: Essential

Relationships:

- Association: Lecturer associates with the CBT system to upload questions.

Include:

- Validate CSV File:

- Lecturer navigates to the "Upload CSV Questions" tab.

- Lecturer selects a CSV file containing the questions.

- System validates the file format and structure.

- Verify Duplicate Questions:

- System checks for the presence of duplicate questions in the uploaded file.

- An error message is displayed if duplicates are found.

- Upload Questions:

- System uploads the questions from the CSV file.

- Confirmation message is displayed upon successful upload.

- Questions are added to the question bank.

Extend:

- Error Handling:

- In case of issues during the validation or upload process, the system presents an error message to the lecturer.

Generalization: None

Normal Flow of Events:

Lecturer accesses the dedicated tab to specify the desired number of questions for the test.

The system validates the entered number, displaying a success message for a valid entry and an error message for an invalid one.

CSV File Upload:

Lecturer can upload questions in a CSV file.

The application verifies the absence of duplicate questions in the uploaded file.

If successful, a confirmation message is displayed, indicating the creation of the question bank.

In case of duplicates or other issues, an error message is presented.

Alternate Flow: Invalid Number of Questions:

If the entered number of questions is invalid:

The system displays an error message indicating "Invalid number of questions."

Alternate Flow: Duplicate Questions in CSV File:

If the uploaded CSV file contains duplicate questions:

The system displays an error message indicating "Duplicate questions found in the file."

Asks the lecturer to correct the CSV file

Use Case Name: Set Timer

ID: UC-ST

Primary Actor: Lecturer

Use Case Type: Detail, Essential

Stakeholders and Interests:

- Lecturer: Wants to set the time duration for a test within the Computer-Based Testing (CBT) application.

- System Administrator: Manages the overall functionality and security of the CBT application.

Brief Description:

This use case outlines the process by which a lecturer sets the time duration for a test within the Computer-Based Testing (CBT) application. The duration is a crucial parameter in controlling the test-taking process.

Trigger: Lecturer wants to define the time limit for a specific test.

Importance Level: High

Type: Essential

Relationships:

- Association: Lecturer associates with the CBT system to configure test settings.

Include:

- Login:

- Lecturer logs into the CBT application using valid login credentials.

- System authenticates the lecturer's information and provides access to the lecturer's dashboard.

- Configure Test Duration:

- Lecturer navigates to a dedicated tab for setting the time duration.

- Lecturer inputs the desired duration in hours and minutes.

- Upon clicking the "Submit" button, the system verifies the validity of the entered duration.

- Display Confirmation or Error Message:

- If the entered duration is valid, a success message is displayed.

- If the entered duration is invalid, an error message informs the lecturer of the failure.

Extend: None

Generalization: None

Normal Flow of Events:

Login:

Lecturer logs into the CBT application using valid login credentials.

System authenticates the lecturer's information and provides access to the lecturer's dashboard.

Configure Test Duration:

Lecturer navigates to a dedicated tab for setting the time duration.

Lecturer inputs the desired duration in hours and minutes.

Upon clicking the "Submit" button, the system verifies the validity of the entered duration.

Alternate Flow: Invalid Duration:

If the entered duration is invalid:

The system displays an error message indicating "Invalid test duration."

Use Case Name: Delete Question

ID: UC-DQ

Primary Actor: Lecturer

Use Case Type: Detail, Essential

Stakeholders and Interests:

- Lecturer: Wants to manage and maintain the question bank efficiently.

- System Administrator: Ensures the security and overall functionality of the Computer-Based Testing (CBT) application.

Brief Description:

This use case outlines the process by which a lecturer deletes a question from the question bank within the CBT application. Lecturers can remove questions to maintain the quality and relevance of the test content.

Trigger: Lecturer decides to delete a question.

Importance Level: High

Type: Essential

Relationships:

- Association: Lecturer associates with the CBT system to manage the question bank.

Include:

- Search Question:

- Lecturer navigates to the "Delete Question" tab.

- Inputs the question details or identifier and clicks the "Search" button.

- Validate Question:

- System validates the question details and retrieves the corresponding question information.

- In case of invalidation, the system displays a message indicating "Could Not Find Question."

- Display Question Details:

- For a valid question, the system displays the question details to the lecturer.

- Confirm Deletion:

- Lecturer confirms the deletion by clicking the "Delete" button.

- The system re-verifies the deletion request.

- Display Confirmation:

- Upon successful deletion, the system displays a confirmation message indicating "Question Deleted."

- Error Handling:

- In case of any issues during the deletion process, the application presents an error message to the lecturer.

Extend: None

Generalization: None

Normal Flow of Events:

Search Question:

Lecturer navigates to the "Delete Question" tab.

Inputs the question details or identifier and clicks the "Search" button.

Validate Question:

System validates the question details and retrieves the corresponding question information.

In case of invalidation, the system displays a message indicating "Could Not Find Question."

Display Question Details:

For a valid question, the system displays the question details to the lecturer.

Confirm Deletion:

Lecturer confirms the deletion by clicking the "Delete" button.

The system re-verifies the deletion request.

Use Case Name: View Students’ Scores

ID: UC-VSS

Primary Actor: Lecturer

Use Case Type: Detail, Essential

Stakeholders and Interests:

- Lecturer: Wants to access and review the scores of all students who took a test.

- System Administrator: Manages the overall functionality and security of the CBT application.

Brief Description:

This use case describes how a lecturer can view the scores of all students who have taken a test within the Computer-Based Testing (CBT) application. The lecturer has the option to save the scores to a CSV file.

Trigger: Lecturer wants to access and review student scores.

Importance Level: High

Type: Essential

Relationships:

- Association: Lecturer associates with the CBT system to view student scores.

Include:

- Access Scores:

- Lecturer logs into the CBT application using valid login credentials.

- The system authenticates the lecturer's information and provides access to the dashboard.

- View Scores:

- Lecturer clicks on the "View" button to access the scores of all students.

- The system displays the scores in a tabular format.

- Save Scores to CSV:

- Lecturer has the option to save the scores of the students to a CSV file.

- Clicking on the "Save to CSV" button initiates the process.

Extend: None

Generalization: None

Use Case Name: Modify Student Records

ID: UC-MSR

Primary Actor: Lecturer

Use Case Type: Detail, Essential

Stakeholders and Interests:

- Lecturer: Wants to efficiently update and maintain accurate student records.

- System Administrator: Manages the overall functionality and security of the Computer-Based Testing (CBT) application.

Brief Description:

This use case outlines the process by which a lecturer modifies student records within the Computer-Based Testing (CBT) application. The lecturer searches for a student using their matriculation number, retrieves and displays student details, and updates the records as necessary.

Trigger: Lecturer wants to modify a student's information.

Importance Level: High

Type: Essential

Relationships:

- Association: Lecturer associates with the CBT system to modify student records.

Include:

- Search Student Record:

- Lecturer clicks on the "Update Student Record" tab.

- Inputs the student matriculation number and clicks the "Search" button.

- The system validates the matriculation number.

- Display Student Details:

- If the matriculation number is valid, the system retrieves and displays the student details.

- If invalid, an error message indicates "Could Not Find Student."

- Update Student Record:

- After making necessary changes, lecturer clicks the "Update" button.

- The system re-verifies the student details.

- A confirmation message is displayed, signifying "Record Updated."

- Error Handling:

- In case of any issues during the update process, the application promptly presents an error message.

Extend: None

Generalization: None

Normal Flow of Events:

Search Student Record:

Lecturer clicks on the "Update Student Record" tab.

Inputs the student matriculation number and clicks the "Search" button.

The system validates the matriculation number.

Display Student Details:

If the matriculation number is valid, the system retrieves and displays the student details.

If invalid, an error message indicates "Could Not Find Student."

Update Student Record:

After making necessary changes, lecturer clicks the "Update" button.

The system re-verifies the student details.

A confirmation message is displayed, signifying "Record Updated."

Error Handling:

In case of any issues during the update process, the application promptly presents an error message.

Alternate Flow: Cancel Update:

If the lecturer decides not to update the student record:

The system cancels the update process and returns to the student details.

Use Case Name: Register Fellow Lecturers

ID: UC-RFL

Primary Actor: Lecturer (Registrar)

Use Case Type: Detail, Essential

Stakeholders and Interests:

- Lecturer (Registrar): Wants to efficiently register fellow lecturers within the Computer-Based Testing (CBT) application.

- System Administrator: Manages the overall functionality and security of the CBT application.

Brief Description:

This use case outlines the process by which a lecturer, acting as a registrar, registers fellow lecturers within the Computer-Based Testing (CBT) application. The registrar can register fellow lecturers individually or in bulk.

Trigger: Registrar wants to add fellow lecturers to the system.

Importance Level: High

Type: Essential

Relationships:

- Association: Lecturer (Registrar) associates with the CBT system to register fellow lecturers.

Include:

- Individual Lecturer Registration:

- Registrar supplies necessary information for individual lecturer registration.

- Staff ID undergoes validation.

- Random password generated and sent via email upon successful validation.

- Confirmation message displayed for successful record creation.

- Error message displayed in case of validation failure or other errors.

- Bulk Lecturer Registration:

- Registrar uploads lecturer records in a CSV file.

- Staff IDs are verified.

- Random passwords generated and sent via email for valid entries.

- Confirmation message displayed for successful record creation.

- System refrains from generating passwords or sending emails for invalid entries.

Extend: None

Generalization: None

Use Case Name: Expunge Student Record

ID: UC-ESR

Primary Actor: Lecturer

Use Case Type: Detail, Essential

Stakeholders and Interests:

- Lecturer: Wants to remove a student's record from the Computer-Based Testing (CBT) application.

- System Administrator: Manages the overall functionality and security of the CBT application.

Brief Description:

This use case outlines the process by which a lecturer removes a student's record from the Computer-Based Testing (CBT) application. The lecturer searches for a student, validates the information, and initiates the deletion process.

Trigger: Lecturer wants to expunge a student's record.

Importance Level: High

Type: Essential

Relationships:

- Association: Lecturer associates with the CBT system to expunge a student's record.

Include:

- Search Student Record:

- Lecturer navigates to the "Delete Student Record" tab.

- Inputs the student's matriculation number and clicks the "Search" button.

- Validate Matriculation Number:

- System validates the provided matriculation number.

- Displays a message indicating "Could Not Find Student" in case of invalidation.

- Retrieves and displays student details in case of validation.

- Confirm Deletion:

- After reviewing the student details, the lecturer confirms the deletion by clicking the "Delete" button.

- The system re-verifies the deletion request.

- Display Confirmation:

- If the deletion is successful, a confirmation message is displayed, signifying "Record Deleted."

- In case of any issues during the deletion process, an error message is presented.

Extend: None

Generalization: None

Normal Flow of Events:

Individual Lecturer Registration:

Registrar supplies necessary information for individual lecturer registration.

Staff ID undergoes validation.

Random password generated and sent via email upon successful validation.

Confirmation message displayed for successful record creation.

Error message displayed in case of validation failure or other errors.

Bulk Lecturer Registration:

Registrar uploads lecturer records in a CSV file.

Staff IDs are verified.

Random passwords generated and sent via email for valid entries.

Confirmation message displayed for successful record creation.

System refrains from generating passwords or sending emails for invalid entries.

Alternate Flow: Existing Account:

If the registrar tries to register a lecturer with an existing account:

The system displays an error message indicating "Lecturer already registered."