StudyBuddy
Use Cases and Test Cases

2005001 Anik Saha[1] 2005006 Kowshik Saha[2] 2005012 Abrar Jahin Sarker[3] 2005013 Al Muhit Muhtadi[4] 2005017 Abdullah Muhammed Amimul Ehsan[5] 2005023 Jaber Ahmed Deedar[6]

Slides prepared and presented by 2005017 Abdullah Muhammed Amimul Ehsan

[1,2,3,4,5,6]Department of Computer Science and Technology, BUET

October 29, 2024

Use Case Overview: Resource Document Preprocessing

Overview and Steps for Resource Document Preprocessing

Scenario: System processes an uploaded PDF document to extract various components including text, images, tables, and generates QA pairs
Preconditions: Operator has a valid PDF document to upload
Actors: Operator, System
Steps:

1. Operator uploads a PDF document
2. System validates the document format and size
3. System extracts components from the document
4. System generates initial parsing results

Test Cases for Resource Document Preprocessing

Test Case Name	**Description**
Valid PDF Processing	Verify that the system can successfully process a valid PDF document and extract components
Invalid File Format	Verify that the system properly handles non-PDF file uploads
Oversized File	Verify that the system handles PDF files exceeding size limit

Use Case Overview: Content Chunking

Overview and Steps for Content Chunking

Scenario: System breaks down processed text content into manageable chunks for further processing
Preconditions: Valid text content is available in text or markdown format
Actors: System
Steps:

1. System receives parsed text content
2. System validates the input format
3. System chunks the content based on specified size
4. System returns the chunked content

Test Cases for Content Chunking

Test Case Name	**Description**
Valid Text Chunking	Verify that the system can successfully chunk valid text content
Empty Content	Verify system handling of empty text content
Unsupported Format	Verify system handling of unsupported file formats

Use Case Overview: Content Vectorization

Overview and Steps for Content Vectorization

Scenario: System vectorizes chunked content and stores it in vector database
Preconditions: Valid chunked content is available with proper metadata
Actors: System
Steps:

1. System receives chunked content with metadata
2. System validates chunk sizes and metadata
3. System vectorizes the content
4. System stores vectors in specified database

Test Cases for Content Vectorization

Test Case Name	Description
Valid Vectorization	Verify successful vectorization and storage of valid content chunks
Token Limit Exceeded	Verify handling of chunks exceeding embedding model token limit
Invalid Metadata	Verify handling of invalid metadata types

Use Case Overview: QA Generation

Overview and Steps for QA Generation

Scenario: System processes and stores generated question-answer pairs
Preconditions: Valid QA pairs are generated from processed content
Actors: System, Moderator
Steps:

1. System receives generated QA pairs
2. System validates QA structure and content
3. System categorizes questions by type
4. System stores finalized QA pairs

Test Cases for QA Generation

Test Case Name	**Description**
Valid Theoretical QA	Verify processing of valid theoretical question-answer pairs
Valid Problem Solving QA	Verify processing of valid problem-solving question-answer pairs
Inconsistent Problem Solving QA	Verify processing of inconsistent problem-solving question-answer pairs

Use Case Overview: Handling Inappropriate Messages in Conversation

Overview and Steps for Handling Inappropriate Messages in Conversation

Scenario: The system detects inappropriate messages and responds appropriately by flagging or rejecting the content.
Preconditions: Student is engaged in conversation with the System.
Actors: Student, System
Steps:

1. Student sends an inappropriate message (e.g., offensive, harmful).
2. System detects inappropriate content using content moderation techniques.
3. System responds with a warning and stops further interaction.

Test Cases for Handling Inappropriate Messages in Conversation

Test Case Name	Description
Appropriate Message Interaction	Verify that the System allows a polite, non-harmful message to pass without warnings or interruptions.
Detect Inappropriate and Harmful Content	Verify that the System detects and flags inappropriate content like offensive language ,harmful suggestions or violent actions.
Confidential Information Request	Verify that the System refuses to provide or request private/confidential information.

Use Case Overview: Student Input Validity for Files

Overview and Steps for Student Input Validity for Files

Scenario: The system validates Student input, ensuring that the correct file types, sizes, and formats are accepted.
Preconditions: Student uploads a file for processing.
Actors: Student, System
Steps:

1. Student attempts to upload a file.
2. System checks file type (allowed: images,txt).
3. System checks the file size to ensure it does not exceed limits.

Test Cases for Student Input Validity for Files

Test Case Name	**Description**
Valid File Upload	Verify that the system allows uploading valid file types within size limits.
Invalid File Type	Verify that the system rejects unsupported file types.
File Size Exceeds Limit	Verify that the system rejects files that exceed the maximum size limit.

Use Case Overview: Explanation Request

Overview and Steps for Explanation Request

Scenario: The student requests explanations for academic or conceptual questions from various fields, either by asking a question directly or by uploading a document. The student can follow up for further clarifications.
Preconditions: The student asks a question or uploads a document that requires an explanation related to a concept or a problem or a solution.
Actors: Student, System
Steps:

1. Student asks a valid academic or conceptual question or uploads a document for analysis.
2. System retrieves relevant information from its knowledge base or analyzes the uploaded document.
3. System provides a clear and concise explanation.
4. System allows the student to ask follow-up questions for further clarification.

Test Cases for Explanation Request

Test Case Name	Description
Valid Explanation Request	Verify that the System explains the logic behind a specific concept,problem or solution.
Follow-Up Question for Deeper Explanation	Verify that the system allows follow-up questions for deeper understanding of a problem.
Public Document Selection for Problem Explanation	Verify that the system analyzes a problem presented in a selected public document and provides a relevant response.
Handling Complex Problem with Step-by-Step Hints	Verify that the system provides step-by-step hints for complex problems instead of disclosing the full solution at once.
Ambiguous Question	Verify that the system asks for clarification when the student provides an incomplete or vague question.
No Relevant Info in Database	Verify that the system handles cases where no relevant information is found in the database.

Use Case Overview: Content Generation Request

Overview and Steps for Content Generation Request

Scenario: Student requests the system to generate content such as flashcards, slides, cheatsheets, or notes.
Preconditions: Student has requested specific content generation (e.g., flashcards).
Actors: Student, System
Steps:

1. Student requests content (e.g., flashcards, slides).
2. System generates requested content with default options
3. Student may or may not provide customization options
4. System makes necessary changes
5. System provides the generated content to the Student.

Test Cases for Content Generation Request

Test Case Name	**Description**
Valid Content Request	Verify that the system generates the requested content (e.g., flashcards or notes) and customizes the format.
Request for Invalid Content Type	Verify that the system handles invalid content types by rejecting the request.
Request for Customization Beyond Limits	Verify that the system handles unrealistic customization requests such as very high resolution for small files.

Use Case Overview: User Interaction and Usability: Understanding and Correcting Input

Overview and Steps for User Interaction and Usability: Understanding and Correcting Input

Scenario: The system continues to function regardless of how the student chats by properly understanding and correcting various input styles, including grammatical mistakes, slang, formality, and incomplete sentences.
Preconditions: Student interacts with the system using various types of input, including grammatical errors, slang, incomplete sentences, or colloquial language.
Actors: Student, System
Steps:

1. Student sends a message with grammatical errors, slang, or informal language.
2. System detects and corrects the input, ensuring proper understanding.
3. System provides the appropriate response based on corrected input.

Test Cases for User Interaction and Usability: Understanding and Correcting Input

Test Case Name	**Description**
Handling Varied Communication Styles in a Single Query	Verify that the system understands and responds correctly when a question contains a mix of grammatical errors, slang, overly formal language, and casual/informal language.
Handling Abbreviations and Incomplete Sentences	Verify that the system understands abbreviations and incomplete sentences and provides the correct response.

Use Case Overview: Error Handling for Wrong Solution and Syntax Error Explanation

Overview and Steps for Error Handling for Wrong Solution and Syntax Error Explanation

Scenario: The student asks for explanations of a wrong solution or code with syntax errors, and the system handles these edge cases properly.
Preconditions: Student provides a wrong solution or code with syntax errors and requests an explanation.
Actors: Student, System
Steps:

1. Student asks for an explanation of a wrong solution or code.
2. System detects the anomalies and System explains the correct one.

Test Cases for Error Handling for Wrong Solution and Syntax Error Explanation

Test Case Name	**Description**
Request for Explanation of Wrong Solution	Verify that the system detects a wrong solution provided by the student and explains why the solution is incorrect, followed by the correct solution.
Request for Explanation of Partially Correct Solution	Verify that the system detects when a solution is partially correct and explains which parts are correct and where the error is.
Request for Explanation of Code with Multiple Syntax Errors	Verify that the system handles code with multiple syntax errors by identifying each error and providing feedback for correction.
Request for Explanation of Logical Error in Code	Verify that the system distinguishes between syntax errors and logical errors, explaining the logic mistake in the code.

Use Case Overview: Similar Raw Content Retrieval

Overview and Steps for Similar Raw Content Retrieval

Scenario: System retrieves similar existing content from RawContentDB based on content type and topic description
Preconditions: RawContentDB contains indexed content of various types
Actors: System
Steps:

1. System receives content type and topic description
2. System validates input parameters
3. System searches for similar content in RawContentDB
4. System filters results based on similarity threshold
5. System returns specified number of retrievals

Test Cases for Similar Raw Content Retrieval

Test Case Name	**Description**
Valid Content Retrieval	Verify successful retrieval of similar content matching type and topic
Empty Content Type	Verify system handling of empty content type
Empty Topic	Verify system handling of empty topic description
Type Mismatch With Similar Topic	Verify handling of cases where similar topic exists but in different content type

Use Case Overview: Content Modification

Overview and Steps for Content Modification

Scenario: System modifies existing similar content to match required content type and topic
Preconditions: Similar content has been retrieved from RawContentDB
Actors: System
Steps:

1. System analyzes existing content characteristics
2. System determines required modifications
3. System applies modifications to match required type and topic
4. System validates modified content

Test Cases for Content Modification

Test Case Name	Description
Valid Content Modification	Verify successful modification of content to match required type and topic

Use Case Overview: Resource Retrieval

Overview and Steps for Resource Retrieval

Scenario: System retrieves relevant resources from specified collection based on content topic
Preconditions: Resource collections are properly indexed and available in vector database
Actors: System
Steps:

1. System validates collection name
2. System searches for relevant resources
3. System ranks resources by relevance
4. System returns specified number of resources

Test Cases for Resource Retrieval

Test Case Name	Description
Valid Resource Retrieval	Verify successful retrieval of resources from valid collection
Invalid Collection	Verify handling of non-existent collection
Insufficient Resources	Verify handling of insufficient stored chunks

Use Case Overview: Raw Content Generation

Overview and Steps for Raw Content Generation

Scenario: System generates new raw content from retrieved resources based on specified content type
Preconditions: Relevant resources have been retrieved successfully
Actors: System
Steps:

1. System validates content type support
2. System analyzes retrieved resources
3. System generates content according to type requirements
4. System validates generated content completeness

Test Cases for Raw Content Generation

Test Case Name	**Description**
Valid Content Generation	Verify successful generation of content from adequate resources
Unsupported Content Type	Verify handling of unsupported content type request
Insufficient Content	Verify handling of insufficient content for required type
No Retrieved Resources	Verify handling of empty resource list

Use Case Overview: Quiz Request with Customization Parameters

Overview and Steps for Quiz Request with Customization Parameters

Scenario: Student prompts the system to generate a quiz with customization parameters such as subject, topic, marks, time, and difficulty.
Preconditions: The student is logged in and has access to the quiz generation feature.
Actors: Student, System
Steps:

1. Student requests a quiz through a conversation, providing customization parameters (subject, topic, marks, time, difficulty).
2. System validates the parameters.
3. System proceeds with the quiz generation or provides feedback to the student

Test Cases for Quiz Request with Customization Parameters

Test Case Name	Description
Valid Quiz Request	Verify that a student can successfully request a quiz with valid customization parameters.
Missing Optional Parameters	Verify that the system proceeds with default or recommended values when optional parameters like difficulty are not provided.
Invalid Customization Parameters	Verify that the system handles invalid, unreal or extreme customization parameters such as negative time or unrealistic difficulty.
Missing Required Parameters	Verify that the system asks the student for missing compulsory parameters like subject and topic.
Sensitive Subject or Topic	Verify that the system handles sensitive subjects or topics appropriately by rejecting or sanitizing input.

Use Case Overview: Quiz Question Selection

Overview and Steps for Quiz Question Selection

Scenario: System selects quiz questions based on the customization parameters and student's past performance and peer activity analysis.
Preconditions: The student requests a quiz with valid customization parameters.
Actors: System
Steps:

1. System analyzes the customization parameters as well as past performance and peer activities .
2. System selects a list of selected questions based on the analysis

Test Cases for Quiz Question Selection

Test Case Name	**Description**
Good Distribution of Subtopics and Difficulty	Verify that the selected questions cover all subtopics with a good distribution of difficulty levels.
Lack of Matching Questions	Verify that the system handles the case where there are not enough available questions matching the customization parameters.

Use Case Overview: Quiz Taking Environment

Overview and Steps for Quiz Taking Environment

Scenario: Student answers quiz questions, and the system records the answers immediately.System submits automatically when time runs out.
Preconditions: Quiz is loaded and displayed to the student.
Actors: Student, System
Steps:

1. Student selects answers for each question.
2. System records the answer as soon as the student selects it.

Test Cases for Quiz Taking Environment

Test Case Name	**Description**
Rapidly Changing Answers Before Submission	Verify that the system records the last selected answer when the student rapidly changes answers before submission.
Some Questions Left Blank	Verify that the system allows some questions to be left blank and proceeds with evaluation.
No Answers Provided	Verify that the system prompts the student to review or submit when no answers are provided.
Summary View Display	Verify that the system correctly displays the summary view with the count of answered and unanswered questions, as well as the remaining time.
Answer Recorded Just Before Time Runs Out	Verify that the system records the answer just before time runs out.
Automatic Submission on Timeout	Verify that the system automatically submits the quiz when the timer runs out.

Use Case Overview: Quiz Evaluation and Feedback

Overview and Steps for Quiz Evaluation and Feedback

Scenario: The System matches answers, evaluates the quiz, and provides feedback for incorrect responses.
Preconditions: Student has submitted the quiz, and the answers need to be evaluated.
Actors: System
Steps:

1. System matches submitted answers with the correct ones.
2. System marks the answers as correct or wrong.
3. System provides feedback for incorrect answers.

Test Cases for Quiz Evaluation and Feedback

Test Case Name	Description
Marked Single Correct Option	Verify that the system correctly marks the selected option as either correct or wrong.
Marked Single Wrong Option	Verify that the system correctly marks the selected option as either correct or wrong .
Multiple Correct Answers Allowed	Verify that the system allows and correctly evaluates questions with multiple correct answers.
All Correct Answers Must Be Selected for a Score	Verify that the student must select all correct answers to get a score.
Marking Incorrect Options Along with Correct	Verify that the system correctly handles cases where both correct and incorrect options are selected.

Use Case Overview: Quiz Result and Analytics

Overview and Steps for Quiz Result and Analytics

Scenario: The system shows the quiz result to the student, provides explanations for correct answers, hints for incorrect answers, and updates the student's performance dashboard.
Preconditions: The quiz has been submitted, and the evaluation has been completed.
Actors: Student, System
Steps:

1. System shows the result with correct answers marked in green and incorrect answers marked in red.
2. Student clicks for explanation or hints
3. System provides explanations for correct answers.
4. System provides hints and a brief explanation for incorrect answers.
5. System updates the student's dashboard with strengths, weaknesses, and history of quiz performance.

Test Cases for Quiz Result and Analytics

Test Case Name	**Description**
Result Display with Green for Correct and Red for Incorrect	Verify that the system displays the result with correct answers in green and incorrect answers in red.
Explanation for Correct Answers	Verify that the system provides an explanation for correct answers.
Hint and Brief Explanation for Incorrect Answers	Verify that the system provides a hint and a brief explanation for incorrect answers without giving away the full answer, preserving it for future attempts.

Use Case Overview: Balanced Problem Set Generation

Overview and Steps for Balanced Problem Set Generation

Scenario: System generates a balanced problem set based on topic frequencies and marks distribution.
Preconditions: System has access to topic distribution and required marks.
Actors: System
Steps:

1. System receives topics, frequencies, and marks as input.
2. System validates the input parameters.
3. System calculates the ratio of topics to be included.
4. System generates a problem set based on the ratio and marks distribution.

Test Cases for Balanced Problem Set Generation

Test Case Name	**Description**
Equal Marks with Different Frequencies Distribution	Verify the system generates a problem set with the correct ratio of topics.
Extreme Frequency Distribution	Verify the system handles cases with an extreme frequency distribution.
Extreme Frequency Distribution Considering Obtained Marks	Verify system behavior when one topic has extreme frequency but low marks.

Use Case Overview: Suggest Connections

Overview and Steps for Suggest Connections

Scenario: System suggests connections based on shared institute, subject, and performance in similar quizzes.
Preconditions: Student data includes institute, subjects, and quiz results.
Actors: System
Steps:

1. System receives the student's profile data.
2. System calculates similarities with other students based on weighted criteria.
3. System ranks potential connections by similarity score.
4. System suggests top matching connections.

Test Cases for Suggest Connections

Test Case Name	Description
Same Institute and Subject and Similar Quiz Scores	Verify system prioritizes suggestions from the same institute and subject and similar quiz marks.
Same Institute and Subject	Verify system gives high match scores to suggestions from the same institute and subject.
Different Institute but Same Subject	Verify system suggests students from different institutes with the same subject.
Similar Quiz Marks in Similar Subjects	Verify system handles cases where students have similar marks but different institutes and same subjects.

Use Case Overview: Topic and Content Related Suggestion

Overview and Steps for Topic and Content Related Suggestion

Scenario: System suggests popular content based on topic sequences and feedback.
Preconditions: Content feedback and topic sequences are indexed.
Actors: System
Steps:

1. System receives the student's current content progression.
2. System identifies the next topic in sequence.
3. System searches for popular content related to the next topic or other important subjects.

Test Cases for Topic and Content Related Suggestion

Test Case Name	**Description**
Next Topic in Sequence	Verify system suggests next topic in sequence.
No Topic Started Yet	Verify system suggests a preliminary topic if no topic is started yet.
Important Topics of Subject	Verify system suggests important topics of the subject which has not been started yet.
Popular contents of the Subject	Verify system suggests popular contents of the subject.
No Subject Chosen Yet	Verify system suggests a subject if no subject is chosen yet.

Use Case Overview: Connectivity Status

Overview and Steps for Connectivity Status

Scenario: System provides notifications for connection requests
Preconditions: Student has activity data such as sent connection requests
Actors: System, Student
Steps:

1. System allows sending and tracking connection requests.
2. System updates the dashboard with connection request status.

Test Cases for Connectivity Status

Test Case Name	**Description**
Connection Request Accepted	Verify system correctly updates dashboard when a connection request is accepted.
Connection Request Not Accepted	Verify system updates dashboard when a connection request is not accepted.
Connection Request Ignored	Verify system updates dashboard when a connection request is ignored.

Use Case Overview: Progress Analysis

Overview and Steps for Progress Analysis

Scenario: System provides a detailed analysis of student progress over time.
Preconditions: Student has activity data such as quiz scores, content interactions, and time spent.
Actors: System, Student
Steps:

1. System displays progress curve based on quiz scores and time spent.
2. System allows students to compare progress with connections and global averages.

Test Cases for Progress Analysis

Test Case Name	**Description**
Analyze Progress Over Different Time Durations	Verify system provides accurate progress analysis over various time periods.
Invalid Time Duration for Progress Analysis	Verify system handles invalid time duration input for progress analysis.
Segregate Progress by Topics	Verify system can segregate and display progress by specific topics.