

EdBot

Project Team

Dawood Tanvir 21i-1665

Duaa Fatima 21i-1667

Mahum Raza 21i-1662

Session 2021-2025

Supervised by

Dr. Asif Muhammad



Department of AI/DS

National University of Computer and Emerging Sciences
Islamabad, Pakistan

June, 2025

Contents

Chapter 1.....	0
Introduction	0
1.1 Problem Statement	0
1.2 Scope	0
1.3 Modules.....	1
1.3.1 Module 1: User Authentication	1
1.3.2 Module 2: Content Generation	1
1.3.3 Module 3: Adaptive Assessment Generation	1
1.3.4 Module 4: User Interaction	2
1.3.5 Module 5: Results.....	2
1.3.6 Module 6: Admin Dashboard	2
1.3.7 Module 7: System Management	2
1.4 User Classes and Characteristics	3
Chapter 2.....	4
Project Requirements	4
2.1 Use-case	4
2.2 Extended Use Cases	5
2.2.1 Signup	5
2.2.2 Login	6
2.2.3 Logout.....	7
2.2.4 Search Query	7
2.2.5 Content Generation.....	8
2.2.6 View Slides.....	9
2.2.7 Download Slides	9
2.2.8 Take Assessments	10
2.2.9 Immediate Feedback	11
2.2.10 View Results	11
2.3 Functional Requirements	12

2.3.1	User Access Module	12
2.3.2	User Management Module	13
2.3.3	Content Generation Module	13
2.3.4	Assessment Generation Module	14
2.3.5	Feedback Module	15
Chapter 3.....		18
System Overview.....		18
3.1	Functionality.....	18
3.2	Context	18
3.3	Architectural Design	18
3.4	Design Models.....	20
3.5	Data Design	30

List of Figures

Figure 1: Use case Diagram	4
Figure 2 <i>Domain Model</i>	17
Figure 3 <i>Architectural Design</i>	19
Figure 4 <i>User Authentication sequence diagram</i>	20
Figure 5 <i>Content Generation Sequence Diagram</i>	21
Figure 6 <i>Quiz Generation and Feedback sequence diagram</i>	22
Figure 7 User-access module activity diagram	23
Figure 8 User-management module activity diagram	24
Figure 9 <i>Content-Generation module activity diagram</i>	25
Figure 10 Assessment-Generation module activity diagram	26
Figure 11 <i>Feedback module activity diagram</i>	27
Figure 12 <i>Reporting module activity diagram</i>	28
Figure 13 <i>UML Class diagram</i>	29
Figure 14 JSON-format	30

Chapter 1

Introduction

This proposal outlines the vision, goals, and technical roadmap for building an AI-powered system designed to simplify and automate the creation of educational content and assessments. It explains the problem we're addressing, the tools and techniques we'll employ, and the outcomes we aim to achieve. The goal is to ensure that all stakeholders clearly understand the project's purpose and have a well-defined plan to guide its success from start to finish.

1.1 Problem Statement

Learners often face challenges when they don't receive personalized learning experiences or timely feedback. When assessments aren't tailored to their skill level, they might end up with tasks that are either too easy or too hard, making learning frustrating and less effective. This can leave them without the right balance of challenge and support when they need it most. Delayed feedback only makes things worse, leaving students unsure of their progress and where to improve. As a result, they can feel stuck, leading to even more frustration.

1.2 Scope

The scope of the EDBOT project is to develop an AI-driven educational content generation and assessment system that enhances the learning experience for students and educators. Key functionalities include generating tailored educational content based on user-provided keywords and creating various types of assessments, such as multiple-choice questions, short questions and true/false questions, with difficulty levels adapting to student performance. Additionally, EDBOT will allow users to export content in multiple formats for easy sharing. By leveraging advanced AI technologies, EDBOT aims to streamline content generation and assessment processes, making quality educational resources more accessible to all users.

1.3 Modules

Following are the modules and their functional requirements:

1.3.1 Module 1: User Authentication

This module handles the user onboarding process, allowing users to sign up, log in, and recover their accounts.

1. Signup functionality for user authentication.
2. Login functionality for user authentication.

1.3.2 Module 2: Content Generation

This module enables users to generate educational slides based on specific input keywords. It enhances the content with relevant images and provides export options in multiple formats.

1. Auto-generated slides based on input keywords.
2. Export options in DOCX format.
3. Export options in PDF format.
4. Export options in PPTX format.

1.3.3 Module 3: Adaptive Assessment Generation

This module creates customized assessments for learners. The system adjusts question difficulty based on learner performance and generates different question types to match the content.

1. Generates MCQs.
2. Generates true/false.
3. Generates short answer questions.

1.3.4 Module 4: User Interaction

This module allows users to view, download, and interact with the generated content. It also enables students to take assessments and track their progress.

1. View or download the generated slides.
2. Take quizzes based on the slides.

1.3.5 Module 5: Results

This module tracks the user's performance on assessments and provides detailed feedback. Educators can also view analytics and track student progress.

1. View assessment results and track progress.
2. Analytics dashboard for educators to review student performance.

1.3.6 Module 6: Admin Dashboard

This module provides administrators with tools to manage user accounts, content, and assessments. It also generates reports on system and user performance.

1. User account management.
2. Monitoring of generated content.
3. Monitoring of generated assessments.

1.3.7 Module 7: System Management

This module oversees the overall system performance, including data management and performance tracking, ensuring the smooth operation of the web app.

1. Database management for storing user and content data.
2. System performance monitoring and reporting.

1.4 User Classes and Characteristics

Identify the various user classes that you anticipate will use this product, and describe their pertinent characteristics.

User class	Description
Learner	<p>A Learner interacts with the system to generate personalized educational content, take assessments, and track progress. The system utilizes an LLM (Large Language Model) to generate content slides based on learner input and provides adaptive assessments to test understanding. Feedback is given at the end of assessments, and learners can download both the content and assessments.</p> <p>Number of Users: Thousands of learners, ranging from students to professionals.</p> <p>Usage Frequency: Learners are expected to use the system daily to weekly, depending on learning goals.</p> <p>Device Usage: Desktops, laptops, smartphones, or tablets, both at home and on the go.</p> <p>Key Activities: Sign up and log in to create a personalized experience. Provide prompts to the LLM, which generates content slides with relevant images. Take adaptive assessments, including MCQs, true/false, and short answer questions, based on the content slides. Difficulty levels adjust dynamically based on performance. Receive feedback and explanations for incorrect answers. Download generated slides and assessments in formats like PDF, DOCX, or PPTX.</p>

Chapter 2

Project Requirements

This chapter describes the functional and non-functional requirements of the project.

2.1 Use-case

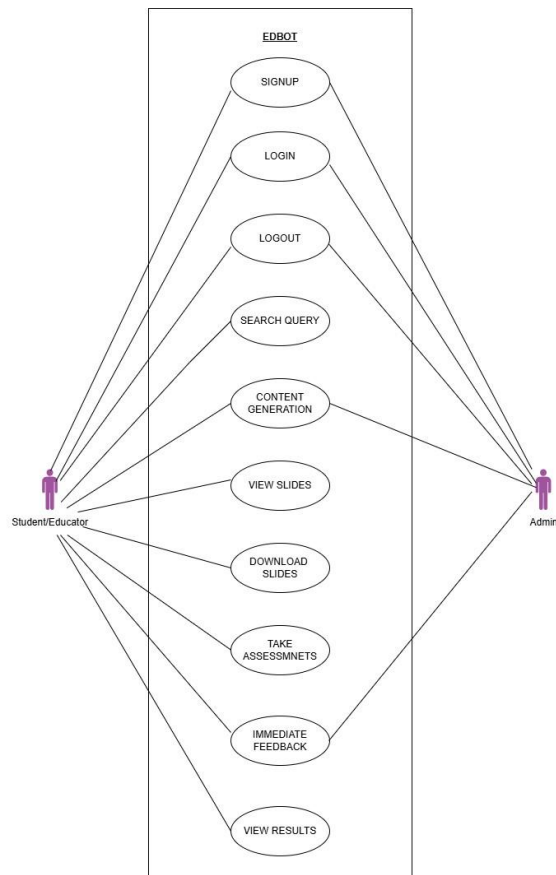


Figure 1: Use case Diagram

2.2 Extended Use Cases

Here are the detailed use cases for EDBOT:

2.2.1 Signup

1. Actor: Learner, Admin
2. Scope: EDBOT Web Application
3. Level: User Goal
4. Preconditions:
 - The user is not already registered on the platform.
 - The signup page is accessible.
5. Success Guarantee:
 - The user is successfully registered on the platform and receives a confirmation email if required.
 - The user can now log in to access their dashboard.
6. Main Success Scenario:
 - The user navigates to the signup page.
 - The user enters personal details such as name, email, and password.
 - The user selects their role as either a student, educator, or admin.
 - The user agrees to the platform's terms and conditions.
 - The system verifies the entered details and creates the user account.
 - The system sends a confirmation email to the user.
 - After email verification, the account becomes active.

7. Extensions:

- The user does not enter mandatory fields (e.g., name, email), and the system prompts for missing information.
- The entered email is already associated with an existing account, and the system asks the user to use a different email or recover their account.

2.2.2 Login

1. Actor: Learner, Admin

2. Scope: EDBOT Web Application

3. Level: User Goal

4. Preconditions:

- The user has a registered account.
- The user knows their login credentials.

5. Success Guarantee: The user is authenticated and successfully logged into the platform.

6. Main Success Scenario:

- The user navigates to the login page.
- The user enters their registered email/username and password.
- The system authenticates the credentials.
- The user is redirected to their respective dashboard based on their role.

7. Extensions:

- The user enters incorrect credentials, and the system displays an error message.
- The user forgets their password and uses the "Forgot Password" option to reset it.

2.2.3 Logout

1. Actor: Learner, Admin
2. Scope: EDBOT Web Application
3. Level: User Goal
4. Preconditions: The user is logged into the system.
5. Success Guarantee: The user is logged out, and their session is securely terminated.
6. Main Success Scenario:
 - The user selects the "Logout" option from the menu.
 - The system terminates the user's session and redirects them to the login page.
 - The user is successfully logged out of the system.
7. Extensions: If the user is inactive for a period, the system automatically logs the user out for security reasons.

2.2.4 Search Query

1. Actor: Learner, Admin
2. Scope: EDBOT Web Application
3. Level: User Goal
4. Preconditions:
 - The user is logged into the system.
 - The search functionality is available and enabled.
5. Success Guarantee: The system successfully retrieves and displays results relevant to the user's query.
6. Main Success Scenario:

- The user enters a search term in the search bar (e.g., specific content, slides, or assessments).
- The system processes the query and searches the database.
- The system returns a list of relevant results.
- The user views or selects from the displayed search results.

7. Extensions:

- If no relevant results are found, the system displays a "No results found" message and suggests related topics.
- . The system may offer filtering or sorting options to refine the search results.

2.2.5 Content Generation

1. Actor: Learner, Admin

2. Scope: EDBOT Web Application

3. Level: User Goal

4. Preconditions:

- The user is logged in and has the necessary permissions to generate content (educator or admin).
- Content creation tools are available.

5. Success Guarantee: The generated content is successfully saved and made accessible to users

6. Main Success Scenario:

- The educator/admin selects the "Create Content" option.
- The user inputs or uploads the content (e.g., lesson plans, slides, videos, assessments).
- The user provides metadata for the content (e.g., title, description, subject, target audience).
- The user reviews the content and submits it for publication.
- The system saves the content and makes it available to students/educators.

- Other users can now access the newly created content.

7. Extensions:

- The user does not complete mandatory fields (e.g., title or subject), and the system prompts them to fill in the missing information.
- The content fails to upload due to a file format or size issue, and the system informs the user of the error.

2.2.6 View Slides

1. Actor: Learner, Admin

2. Scope: EDBOT Web Application

3. Level: User Goal

4. Preconditions:

- The user is logged in.
- Slides are available in the system for the user to view.

5. Success Guarantee: The user can successfully view the selected slides.

6. Main Success Scenario:

- The user selects the option to view slides from the content list.
- The system retrieves the selected slides and displays them to the user.
- The user navigates through the slides.

7. Extensions: If no slides are available, the system displays a message indicating that no content is available.

2.2.7 Download Slides

1. Actor: Learner, Admin

2. Scope: EDBOT Web Application

3. Level: User Goal

4. Preconditions:

- The user is logged in.
- Slides are available for download.

5. Success Guarantee: The user successfully downloads the selected slides.

6. Main Success Scenario:

- The user selects the option to download slides from the content list.
- The system generates a downloadable file for the selected slides.
- The file is downloaded to the user's device.

7. Extensions: If the download fails due to network or file issues, the system prompts the user to retry or troubleshoot the error.

2.2.8 Take Assessments

1. Actor: Student, Educator

2. Scope: EDBOT Web Application

3. Level: User Goal

4. Preconditions:

- The student is logged in.
- Assessments are available for the student to take.

5. Success Guarantee: The student successfully completes and submits the assessment.

6. Main Success Scenario:

- The student selects an available assessment.
- The system loads the assessment for the student.
- The student answers the questions and submits the assessment.
- The system records the student's submission and marks the assessment as completed.

7. Extensions: If the student submits the assessment late, the system warns them about the delay and enforces any penalties (if applicable).

2.2.9 Immediate Feedback

1. Actor: Learner, Admin
2. Scope: EDBOT Web Application
3. Level: User Goal
4. Preconditions:
 - The student has completed an assessment.
 - The system is configured to provide immediate feedback.
5. Success Guarantee: The student receives immediate feedback on their assessment performance.
6. Main Success Scenario:
 - The student completes and submits the assessment.
 - The system evaluates the student's answers.
 - The system provides immediate feedback, showing correct/incorrect answers and explanations (if available).
7. Extensions: If the system cannot evaluate the assessment automatically (e.g., due to manual grading requirements), the feedback is delayed until the educator reviews the submission.

2.2.10 View Results

1. Actor: Learner
2. Scope: EDBOT Web Application
3. Level: User Goal
4. Preconditions:

- The student has completed an assessment or series of assessments.
 - Results are available for viewing.
5. Success Guarantee: The user successfully views their results.
6. Main Success Scenario:
- The student selects the option to view their results.
 - The system retrieves and displays the student's scores, feedback, and analytics.
 - The educator can view the results for individual students or an entire class.
7. Extensions: If the results are not yet available, the system informs the user when they can expect them.

2.3 Functional Requirements

Functional requirements specify the core functionalities and operations that a system must perform to meet user needs. These requirements define how the system should behave, focusing on key tasks such as user interaction, content generation, and assessment handling.

Each module details specific features necessary for the system's overall functionality.

2.3.1 User Access Module

Following are the requirements for module 1:

1. FR1: The system shall display a welcome message to the user upon accessing the application.
2. FR2: The system shall provide navigation options to the login and signup pages from the welcome screen.

3. FR3: The system shall support both guest mode and registered user access.
4. FR4: The system shall offer a brief overview of the EDBOT functionalities and features on the welcome page.

2.3.2 User Management Module

Following are the requirements for module 2:

1. FR5: The system shall allow new users to create an account using an email address and password.
2. FR6: The system shall allow users to log in using previously registered credentials (email and password).
3. FR7: The system shall allow users to recover their password via an email recovery link.
4. FR8: The system shall validate user credentials upon login.
5. FR9: The system shall provide an option to log in with third-party authentication services (e.g., Google, Facebook).
6. FR10: The system shall maintain session management, ensuring user sessions remain active until they log out or the session expires.
7. FR11: The system shall allow users to update their account information, including email and password.

2.3.3 Content Generation Module

Following are the requirements for module 3:

1. FR12: The system shall allow users to input a keyword for which they require content generation.
2. FR13: The system shall fetch and display relevant slides based on the user's subject and topic input.
3. FR14: The system shall allow users to navigate through different slides within a generated slide deck.

4. FR15: The system shall generate educational slides in a structured format, including sections, key points, and examples.
5. FR16: The system shall allow users to download the slides as a PDF for offline use.
6. FR17: The system shall automatically generate relevant images corresponding to the content in the slides to improve the learning experience.
7. FR18: The system shall provide an option for users to mark important slides for later reference.
8. FR19: The system shall auto-save user progress while they are browsing or editing slides.

2.3.4 Assessment Generation Module

Following are requirements for module 4:

1. FR20: The system shall grade assessments based on the topic selected by the user.
2. FR21: The system shall categorize the questions into difficulty levels easy, medium, and hard—depending on the user's performance.
3. FR22: The system shall provide multiple-choice questions (MCQs), true/false, and short answer types according to the content and learner's needs.
4. FR23: The system shall allow users to take timed assessments.
5. FR24: The system shall automatically score the user's responses and include a score report.
6. FR25: The system shall allow users to retry assessments and view their previous attempts and scores.
7. FR26: The system shall summarize assessment results, including areas for improvement.
8. FR27: The system shall dynamically adjust the difficulty level of the assessment questions based on user performance—making questions harder for strong performers and easier for those struggling.
9. FR28: The system shall offer adaptive assessments that become harder or easier based on real-time performance during the session.

10. FR29: The system shall allow users to pause and resume assessments without losing progress.
11. FR30: The system shall allow users to flag questions they find difficult or unclear for review.
12. FR31: The system shall automatically generate relevant and appropriately challenging assessment questions across different difficulty levels.
13. FR32: The system shall use a threshold-based grading system to evaluate student responses and provide personalized, constructive feedback based on performance.

2.3.5 Feedback Module

Following are requirements for module 5:

1. FR33: The system should provide detailed explanations for every question once the assessment is completed.
2. FR34: The system should deliver specific feedback based on the user's performance in the assessment.
3. FR35: The system should allow users to track their progress, helping them understand how they have improved over time in specific areas.
4. FR36: The system shall display performance metrics—such as accuracy, completion time, and progress over multiple assessments—on a visual dashboard.
5. FR37: Immediately after an assessment, the system shall provide AI-driven feedback, enabling learners to quickly understand their performance and areas for improvement.
6. FR38: The system should allow users to submit feedback on questions or assessments they found particularly difficult or confusing.

2.3.6 Reporting Module

Following are requirements for module 6:

1. FR39: The system shall allow users to export their assessment results and feedback in various formats, such as PDF or DOCX, for sharing with peers or instructors.
2. FR40: The system shall generate reports summarizing overall user performance, time spent on topics, and suggested next steps.
3. FR41: The system should log user activity for administrative and analytics purposes to improve system recommendations.

2.4 Non-Functional Requirements

This section specifies nonfunctional requirements.

2.4.1 Testing and test-ability

To test EDBOT as an object-oriented software, we will implement a strategy that includes unit testing, integration testing, and user acceptance testing (UAT). Unit tests will validate the functionality of individual classes and methods, while integration testing will ensure proper interactions between modules. UAT will involve real users to assess usability and functionality, ensuring EDBOT meets user needs. This approach will help identify and resolve issues early, leading to a high-quality educational tool.

2.4.2 Usability

The usability of EDBOT shall be enhanced by a seamless and unintermittent flow for the students as well as the educators. It has to be simple to navigate, instructions were written in plain language and, where possible, there were few things which a user has to learn. Real time, feedback and adaptive assessments shall be readily available and navigable particularly to enhance effective learning and mapping to the system.

2.4.3 Performance

EDBOT will be optimized for quick response times and smooth processing, even when handling large datasets or generating complex educational content. The system will efficiently manage resources to ensure that tasks like content generation, slide

rendering, and assessment feedback are delivered without delays, providing users with a fast and reliable experience.

2.5 Domain Model

Create a representation of the domain model for your project.

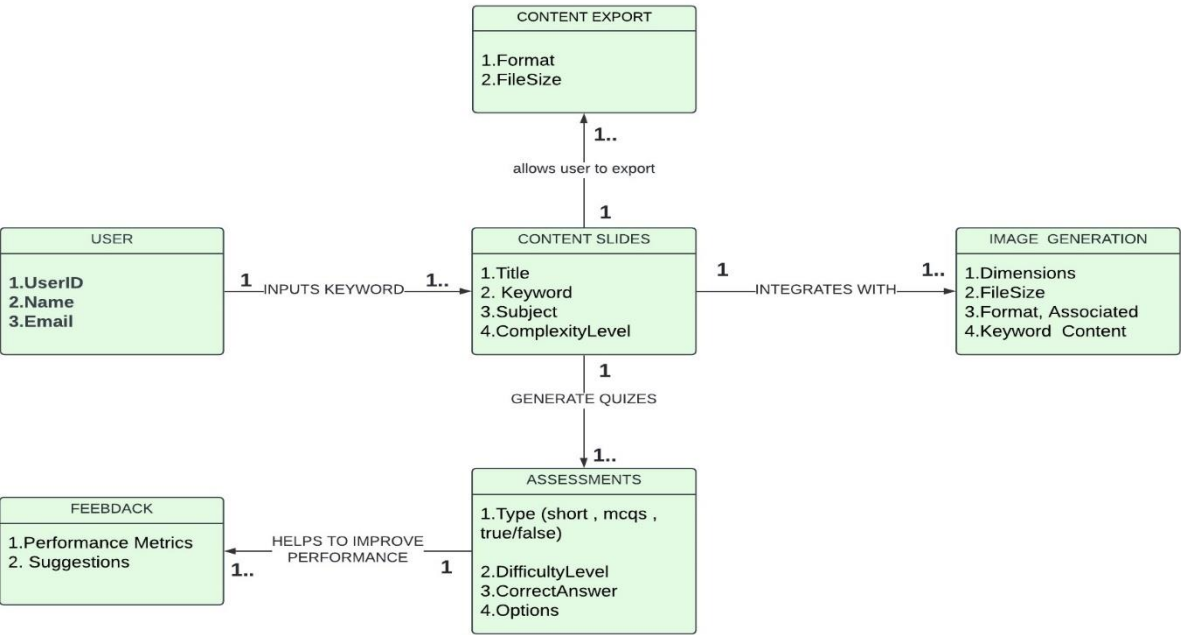


Figure 2 *Domain Model*

Chapter 3

System Overview

3.1 Functionality

The EDBOT project aims to develop an intelligent educational assistant designed to enhance learning experiences through automated content generation and assessment. Users can input keywords or text, prompting EDBOT to generate relevant PowerPoint slides containing educational content and accompanying images. Once users have reviewed the slides, they can take a quiz that consists of multiple-choice questions, short answer questions, and true/false statements. The quiz is timed, and based on the user's performance, EDBOT will adjust the difficulty level for future assessments. After completing the quiz, users receive feedback that includes explanations for any incorrect answers, alongside their overall performance metrics and the opportunity to provide feedback on the quiz itself.

3.2 Context

In today's fast-paced world, efficient learning is paramount, and EDBOT addresses this need by leveraging advanced technologies. The project incorporates Retrieval-Augmented Generation (RAG) techniques to enhance content retrieval and generation, enabling it to deliver personalized educational materials. By automating the creation of learning resources and assessments, EDBOT seeks to improve the efficiency of educational processes and foster a deeper understanding of the subjects studied.

3.3 Architectural Design

The EDBOT system enables users to input text or keywords to generate PowerPoint slides filled with relevant content and images. This process is enhanced by Retrieval-Augmented Generation (RAG), ensuring the material is accurate and comprehensive.

After reviewing the slides, users can take a quiz consisting of multiple-choice, short answer, and true/false questions, all presented in a timed format for added engagement. The system evaluates quiz performance, adjusting the difficulty of future quizzes based on users' mastery of the subject. Detailed explanations for incorrect answers help reinforce learning.

Additionally, users can provide feedback on their quiz experience, aiding in the continuous improvement of EDBOT. Upon completing the quiz, users receive a grade that reflects their performance, giving them insight into their progress and areas for improvement. Overall, EDBOT fosters an adaptive and engaging learning environment through its dynamic integration of RAG technology.

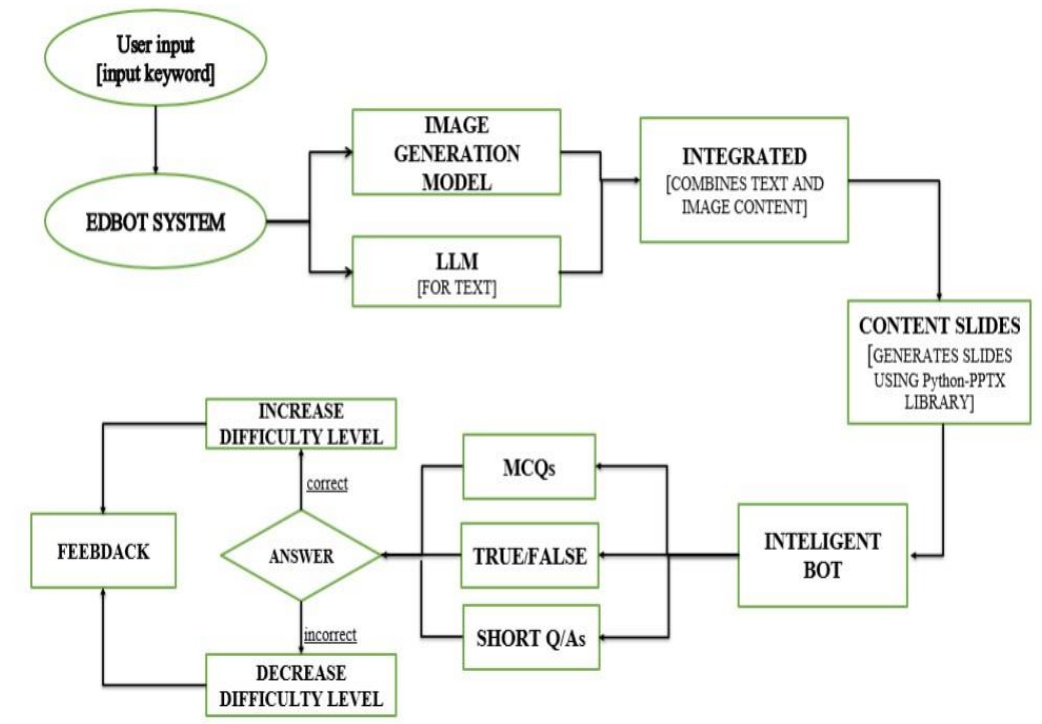


Figure 3 *Architectural Design*

3.4 Design Models

Following are the opted Design models for our project:

3.4.1 Sequence Diagrams

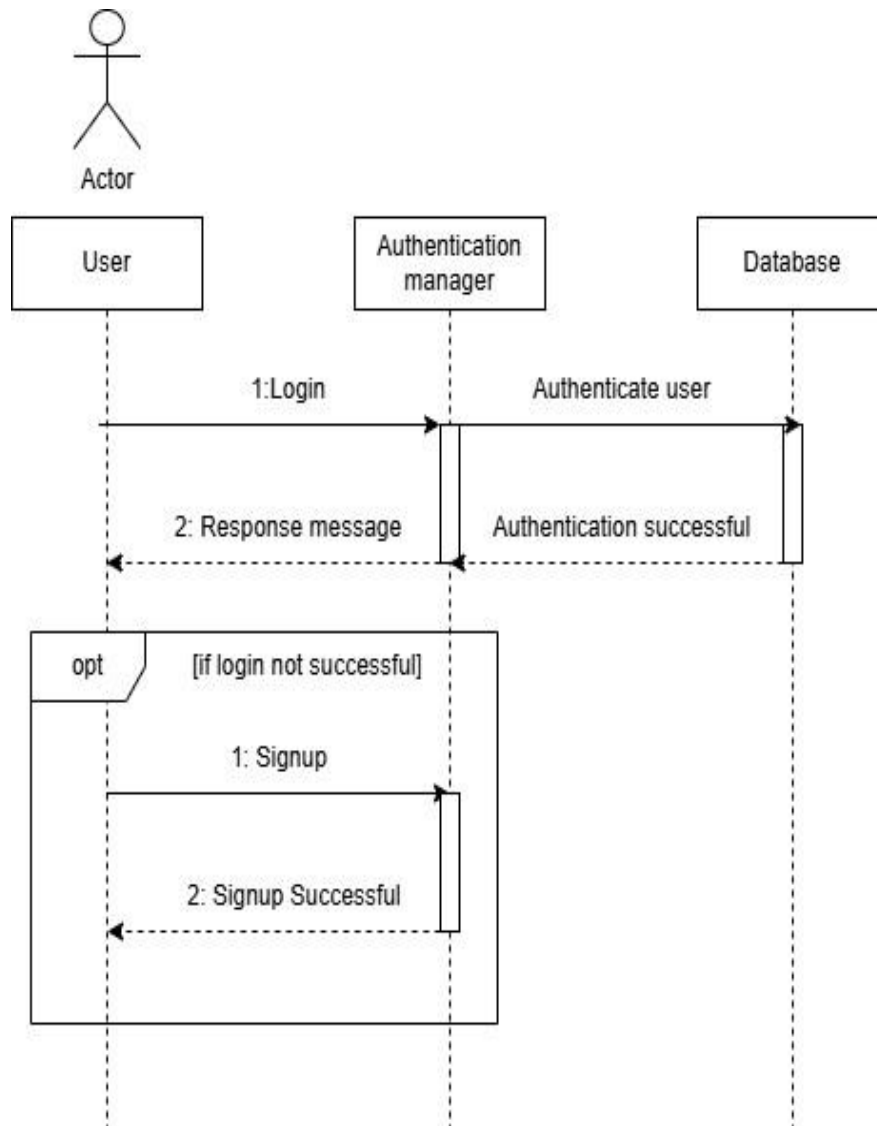


Figure 4 *User Authentication sequence diagram*

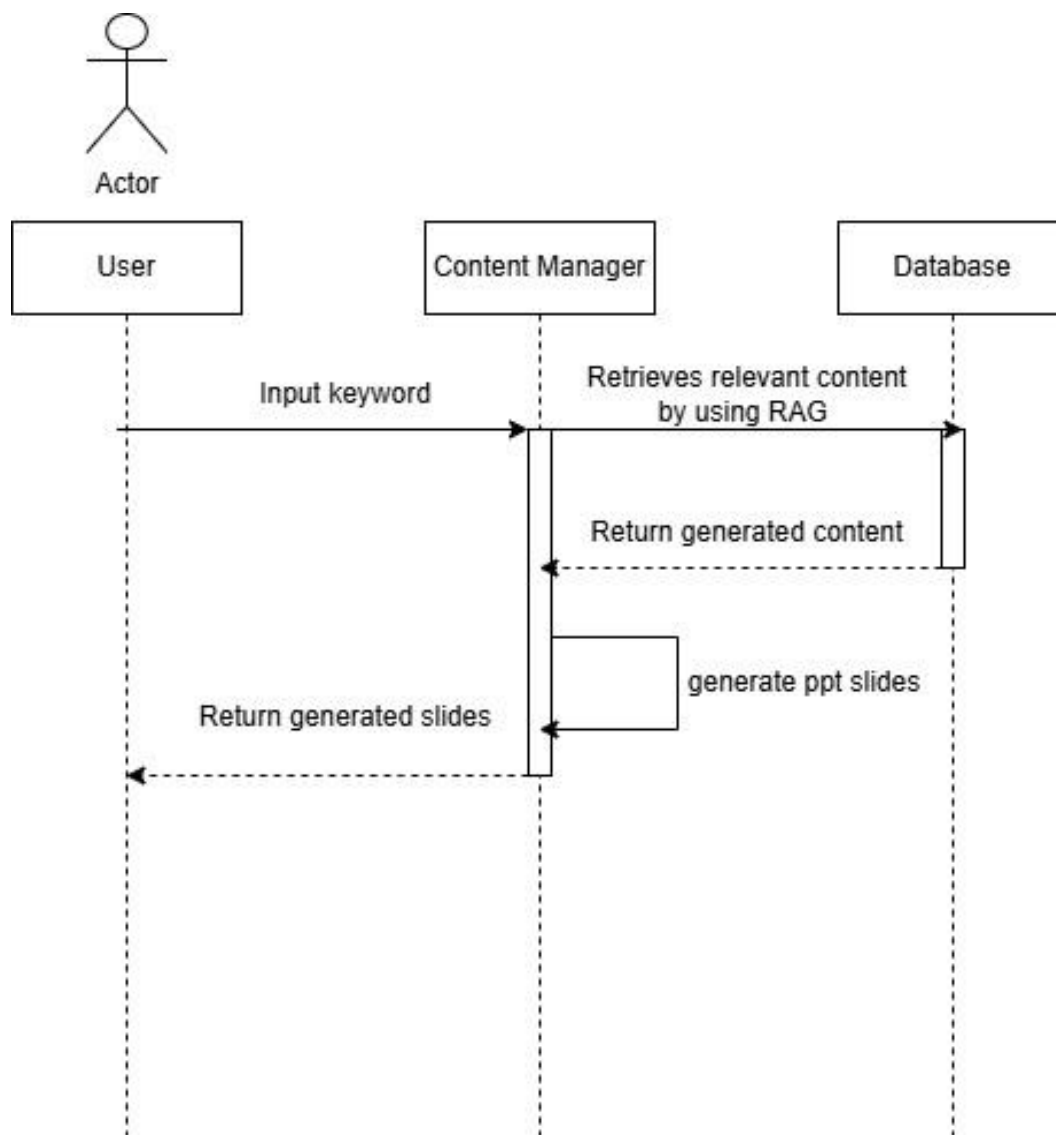


Figure 5 *Content Generation Sequence Diagram*

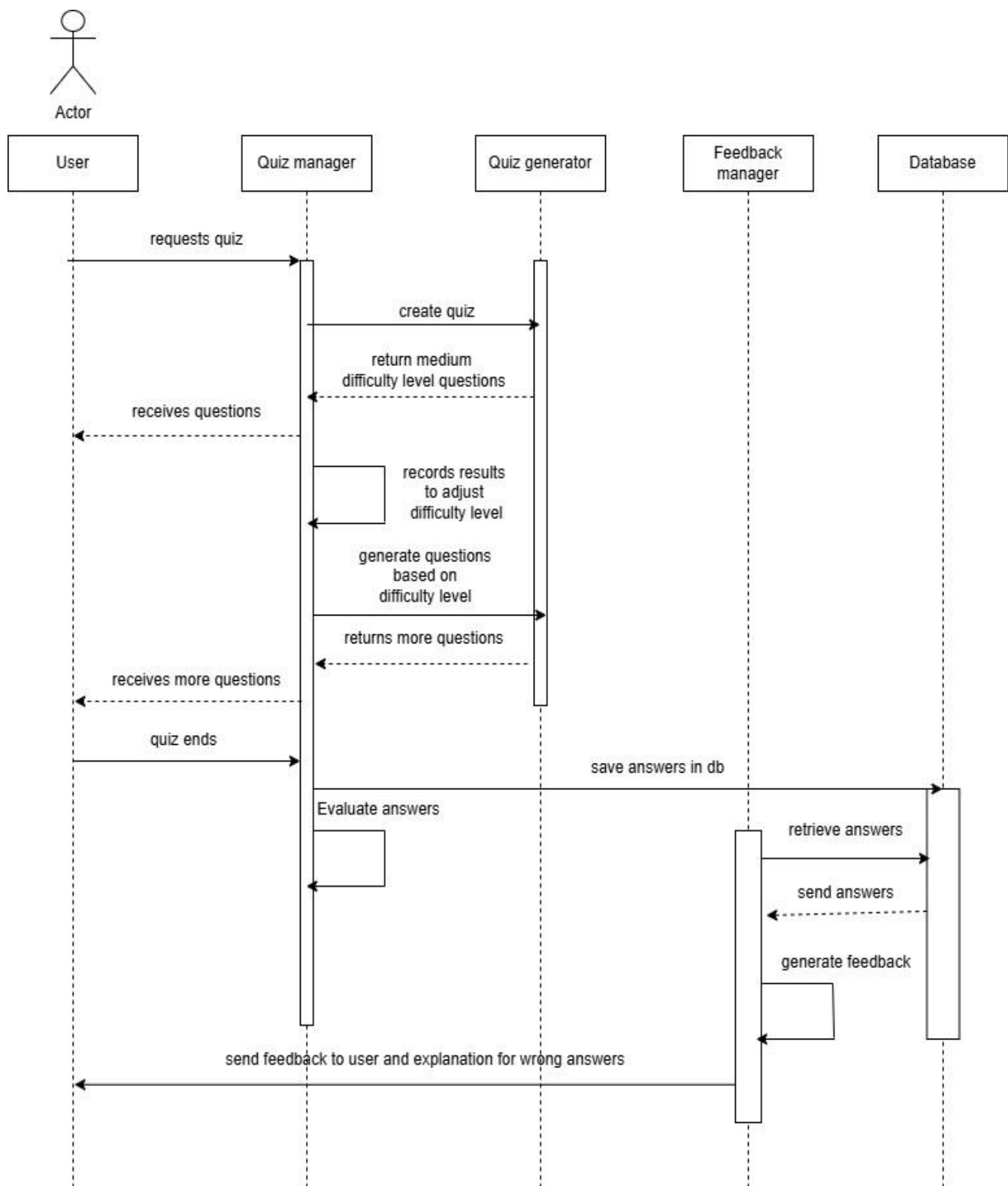


Figure 6 *Quiz Generation and Feedback sequence diagram*

3.4.2 Activity Diagrams

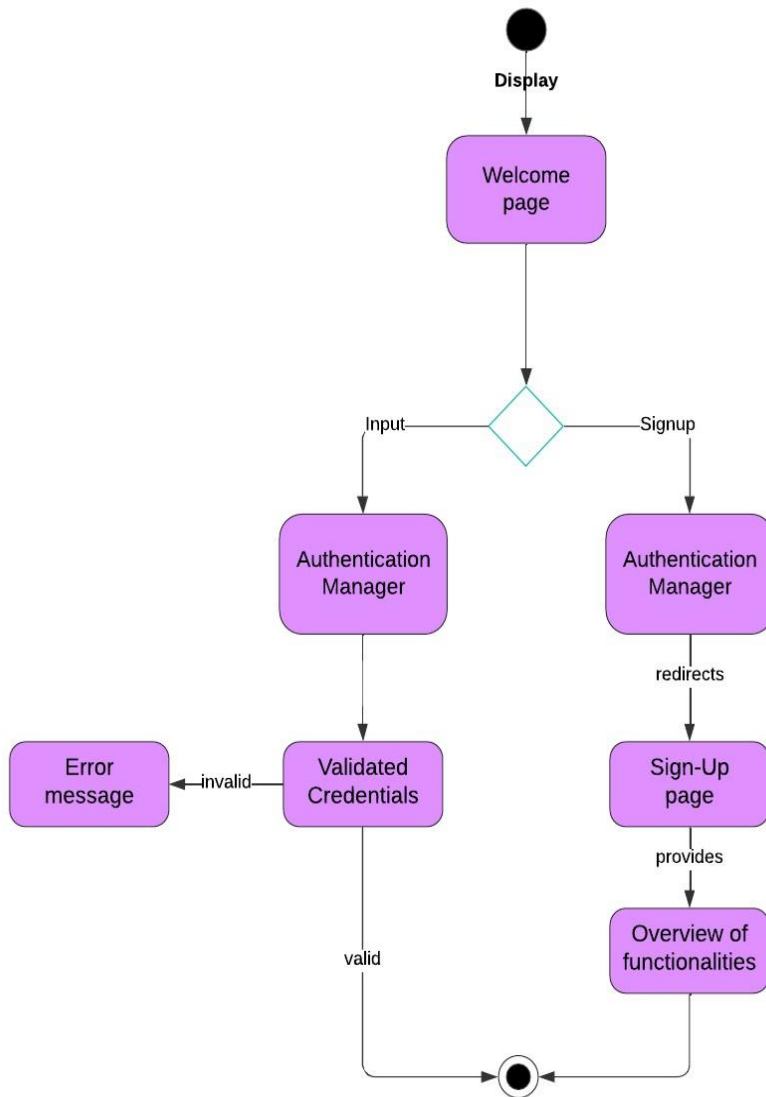


Figure 7 User-access module activity diagram

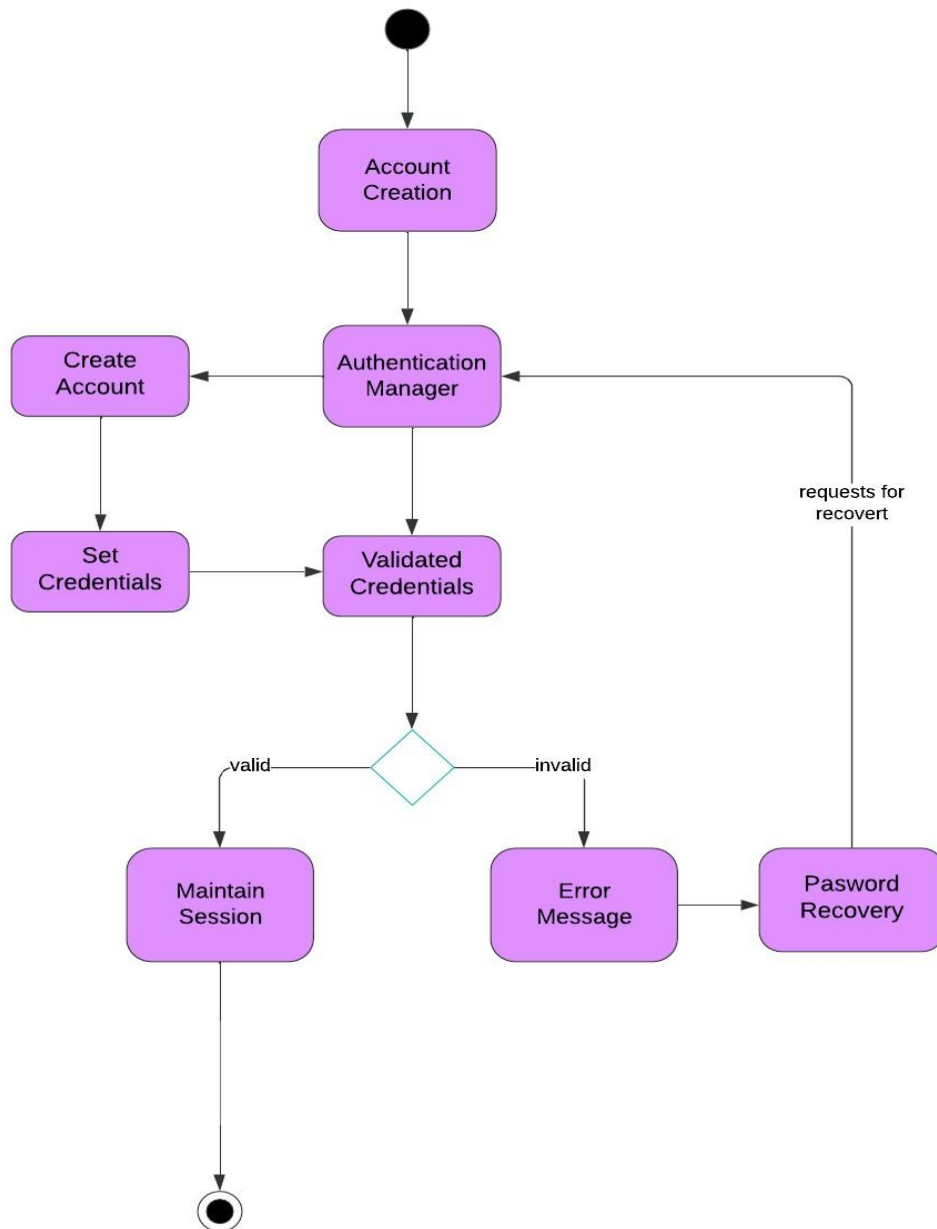


Figure 8 User-management module activity diagram

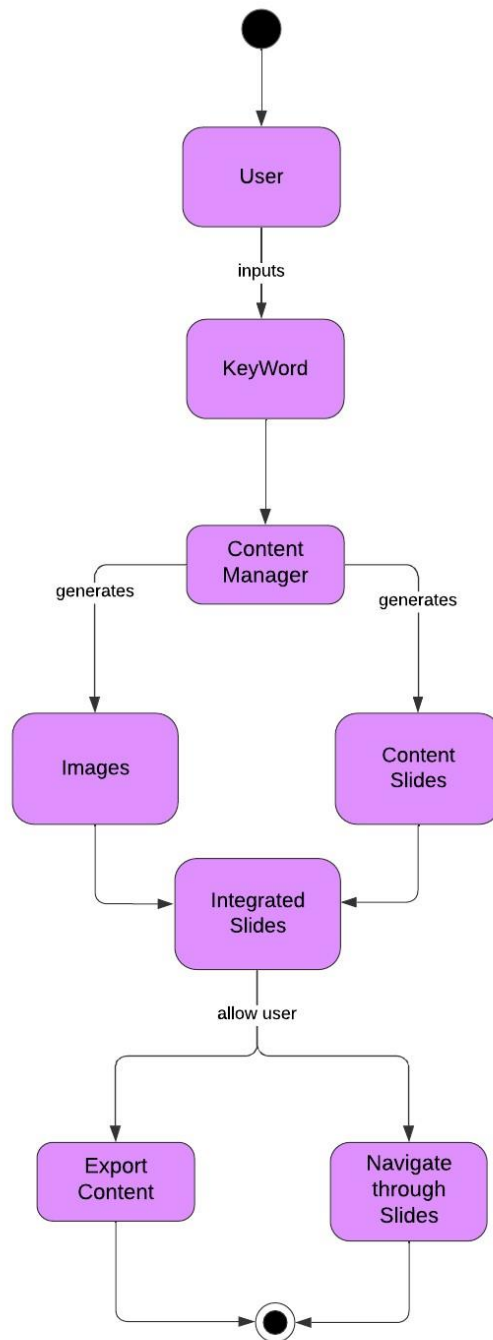


Figure 9 *Content-Generation module activity diagram*

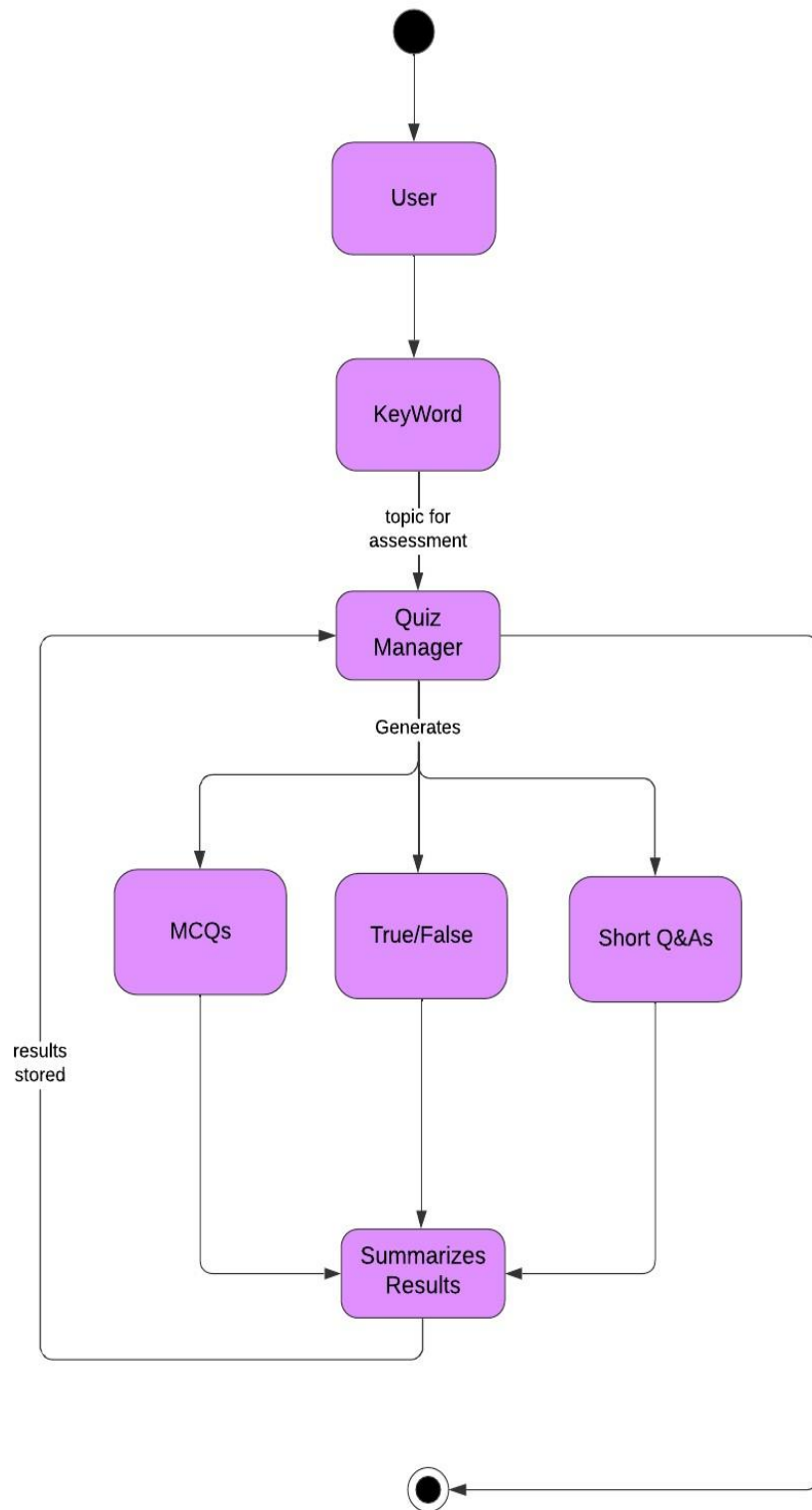


Figure 10 Assessment-Generation module activity diagram

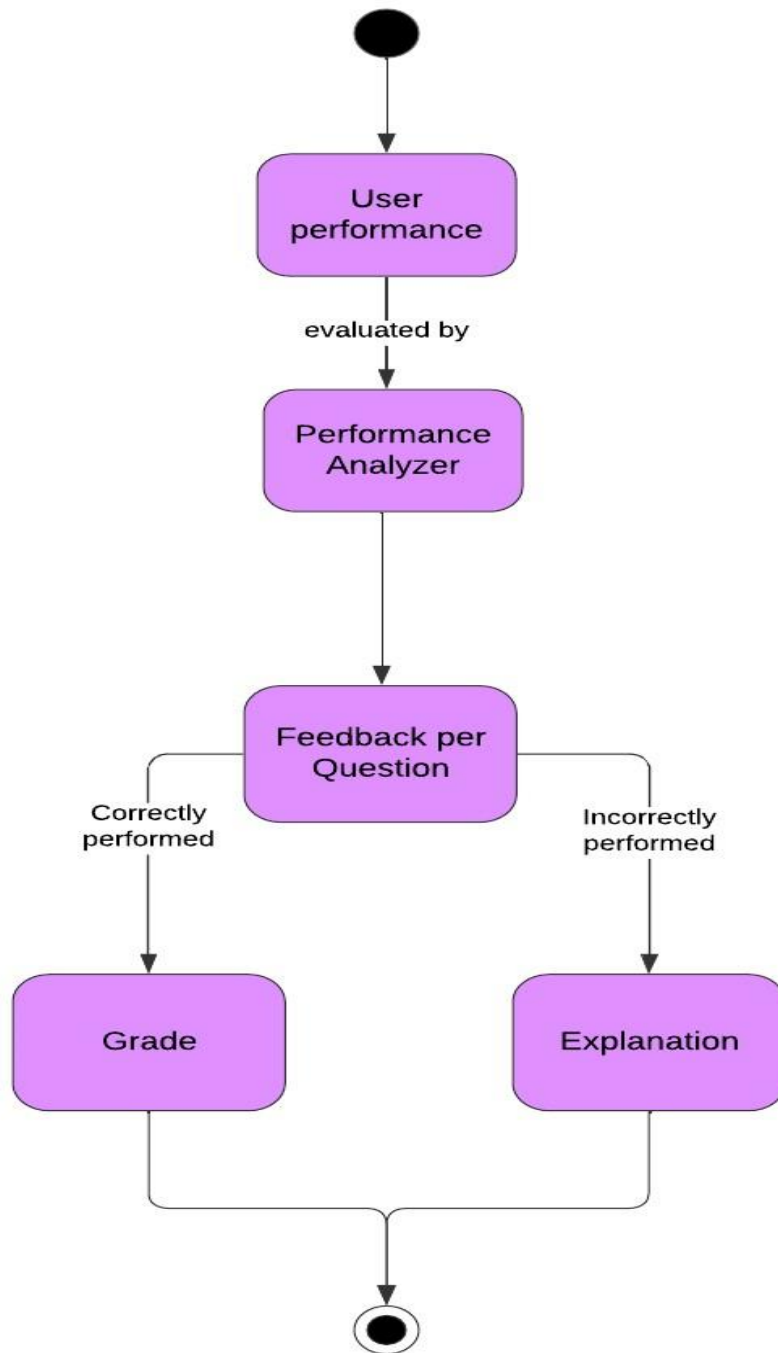


Figure 11 *Feedback module activity diagram*

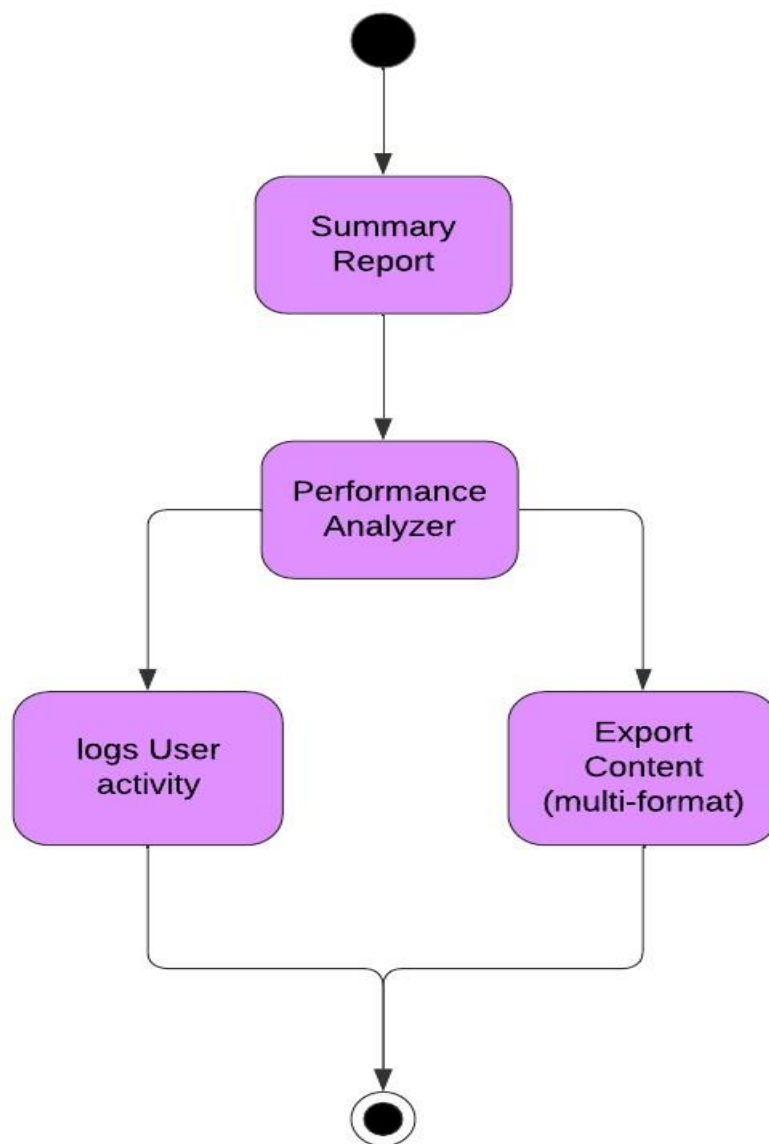


Figure 12 *Reporting module activity diagram*

3.4.3 Class Diagrams

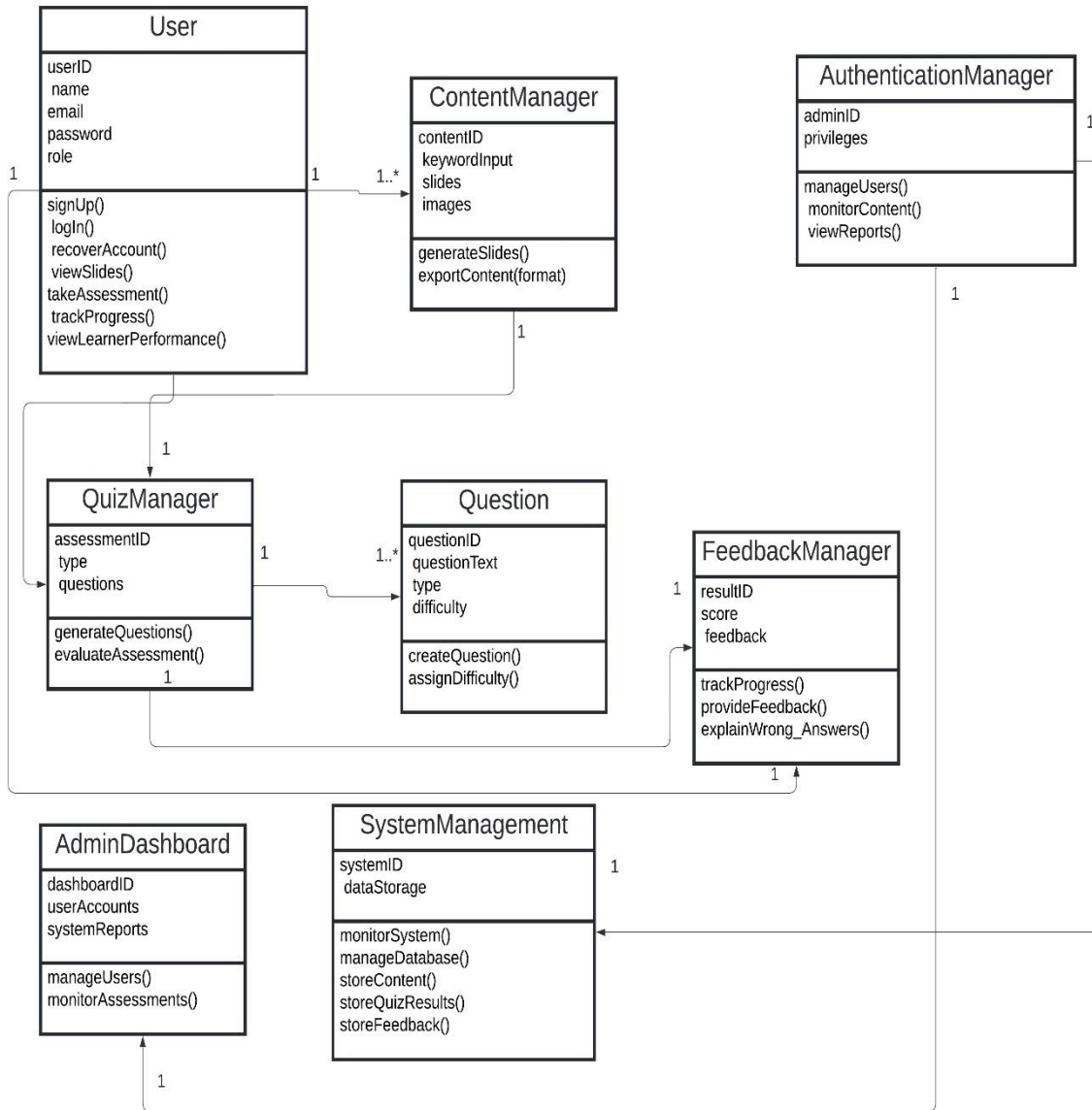


Figure 13 *UML Class diagram*

3.5 Data Design

Overview of the Information Domain This defines the information domain as the raw data and knowledge regarding the subject matter with which the system deals. For our case, this includes:

1. Subject areas (e.g., "Programming Fundamentals")
2. Chapters (e.g., "Introduction to Computers, the Internet, and the Web")
3. Sections (e.g., "1.1 Introduction")
4. Content types (explanation etc.)
5. Content text (detailed information on topics)

These features represent hierarchically and categorically binding information organized in a way that allows efficient access, processing, and retrieval by the system.

```
{
  "EDBOT_data": [
    {
      "subject": "Course_Name",
      "chapters": [
        {
          "chapter": "Chapter_Name",
          "content": [
            {
              "section": "",
              "content_type": "",
              "content_text": ""
            }
          ]
        }
      ]
    }
  ]
}
```

Figure 14 JSON-format