Chapter 3  
This chapter defines the functional and non-functional requirements of the proposed Quiz AI system, along with its constraints and stakeholder interactions. It begins by identifying primary, secondary, and tertiary stakeholders, then outlines detailed software requirements covering authentication, content management, quiz generation, and reporting. Functional requirements are expressed through use cases, while non-functional requirements highlight performance, security, usability, and scalability aspects. The chapter also presents system constraints across resources, technology, operations, and external factors. Finally, software diagrams—including use case, context, and data flow diagrams—illustrate the system’s structure and data interactions, providing a comprehensive view of its expected functionality.

**3.1 Stakeholders**

**3.1.1 Primary Stakeholders**

* **Development Team**: Responsible for building and maintaining the system, thus its success or failure directly impacts them.
* **Students – Group A (Direct Users)**: Actively use the system as a tool for studying and reviewing their learning materials.
* **Teachers**: Benefit from AI-generated quizzes and exams, making assessment preparation faster and easier.
* **College & University Professors**: Similar to teachers, they leverage the AI system to efficiently generate quizzes and exams for higher-level courses.

**3.1.2 Secondary Stakeholders**

* **Students – Group B (Indirect Users)**: While they do not use the system directly, they may still take quizzes and exams created by it.
* **Educational Bodies**: Although not direct users, institutions may need to license or purchase the rights to use the system.

**3.1.3 Tertiary Stakeholders**

* **Textbook Publishers**: May experience reduced demand for premade question banks, negatively impacting sales.
* **Educational Researchers**: Interested in studying how AI usage influences assessment quality and student learning outcomes.
* **Educational Book Sellers & Libraries**: As schools rely more on AI-generated quizzes, demand for printed quiz books and related library resources may decrease.

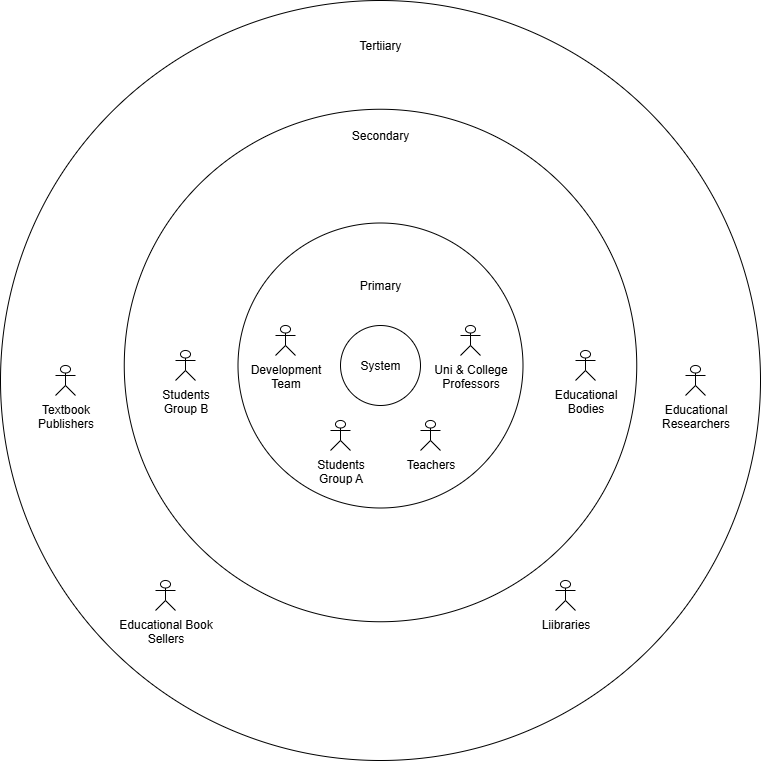
****

Figure 3.1: Stakeholders Diagram for the Quiz AI system, showing the primary, secondary, and tertiary stakeholders.

**3.2 Software Requirements**

1. **User Authentication and Security**
   * Users (students, instructors, admins) must be able to log in securely using their username and password.
   * Users must be automatically logged out if sessions remain idle for too long.
2. **Password Recovery**
   * Users must be able to reset their password through email verification.
   * Users must receive a password reset link or token if they forget their password.
3. **Content Management**
   * Students and instructors must be able to upload content (e.g., lecture slides, PDFs, videos) to generate quizzes.
   * Users must not be able to upload duplicate content files.
4. **Quiz Generation**
   * Users must be able to generate quizzes from uploaded content.
   * Users must be able to choose quiz difficulty level and number of questions.
   * Users must be able to access public quizzes, while private ones remain hidden.
   * Students must be able to view top attempts for public quizzes.
5. **Exam Creation and Publishing**
   * Instructors must be able to create and publish quizzes/exams.
   * Instructors must be able to set attempt limits for each quiz/exam.
6. **Quiz Participation**
   * Students must be able to attempt quizzes/exams within a time limit set by the instructor.
   * Students must have a secure and uninterrupted session during attempts.
7. **Results and Analytics**
   * Students must be able to view their quiz results and weak areas in specific chapters.
   * Instructors must be able to view analytics such as average marks, highest scores, and lowest scores.
8. **Error Handling and Logging**
   * Users must be notified if errors occur during login, uploads, or quiz attempts.
   * The system must ensure critical errors are escalated to administrators.
9. **Administrative Functions**
   * Admins must be able to manage users and their permissions.
   * Admins must be able to generate statistical reports about system usage and performance.

**3.3 Functional User Requirements (Use Case Specifications)**

**1. Login to the System**

**Actor(s):** Student, Instructor, Admin  
**Description:** User enters username and password to access the system.  
**Precondition:** User has an active account.

**Normal Flow:**

1. User navigates to the login page.
2. User enters username and password.
3. System verifies credentials.
4. System grants access to the dashboard.

**Alternate Flow:**

* If credentials are incorrect, the system displays an error message.
* If multiple failed attempts occur, the account is temporarily locked.

**Postcondition:** User is logged in and can access system features.

**2. Handle Login Errors (Acceptance/Denial)**

**Actor(s):** System  
**Description:** System validates credentials and either accepts or denies access.  
**Precondition:** Login request is made by the user.

**Normal Flow:**

1. System checks username and password against the database.
2. If valid, grants access.
3. If invalid, denies access.

**Alternate Flow:**

* System suggests *Forgot Password* after multiple failed attempts.

**Postcondition:** Access is either granted or denied.

**3. Forgot Password Request**

**Actor(s):** Student, Instructor, Admin  
**Description:** User requests password reset through the system.  
**Precondition:** User cannot log in to the system.

**Normal Flow:**

1. User clicks *Forgot Password*.
2. System prompts for registered email address.
3. User enters email address.
4. System sends password reset email.

**Alternate Flow:**

* If email is not registered, the system shows an error message.

**Postcondition:** Password reset email is sent to the user.

**4. Send Password Reset Email**

**Actor(s):** System  
**Description:** Email Control System sends password reset email via Email Service.  
**Precondition:** Password reset request has been made.

**Normal Flow:**

1. System generates password reset token.
2. System sends email with password reset link.

**Alternate Flow:**

* If email service is unavailable, the system logs an error and retries later.

**Postcondition:** User receives password reset email.

**5. Upload Content (Lecture or Exam Material)**

**Actor(s):** Student  
**Description:** Student uploads lecture content for quiz generation.  
**Precondition:** User is logged in and has permission to upload.

**Normal Flow:**

1. User navigates to upload section.
2. User selects content file.
3. System validates file format and size.
4. System stores the content.

**Alternate Flow:**

* If file is invalid, the system displays an error message.

**Postcondition:** Content is uploaded and ready for quiz generation.

**6. Generate Quiz from Uploaded Content**

**Actor(s):** Quiz AI Generation Engine  
**Description:** System processes content and generates a quiz using AI.  
**Precondition:** Content has been successfully uploaded.

**Normal Flow:**

1. System processes uploaded content.
2. AI analyzes content and generates questions.
3. System stores generated quiz.

**Alternate Flow:**

* If processing fails, the system logs an error and notifies the user.

**Postcondition:** Quiz is generated and stored in the system.

**7. View Generated Quiz**

**Actor(s):** Student  
**Description:** Student reviews the AI-generated quiz.  
**Precondition:** Quiz has been generated.

**Normal Flow:**

1. User navigates to quizzes section.
2. User selects generated quiz.
3. System displays the quiz.

**Alternate Flow:**

* If quiz is not available, the system shows an error message.

**Postcondition:** User views the quiz.

**8. Create Public Exam**

**Actor(s):** Instructor  
**Description:** Instructor creates and publishes an exam for public access.  
**Precondition:** Instructor is logged in.

**Normal Flow:**

1. Instructor navigates to exam creation page.
2. Instructor sets exam details and questions.
3. Instructor publishes the exam.

**Alternate Flow:**

* If publishing fails, the system shows an error message.

**Postcondition:** Exam is publicly available.

**9. View Public Exam Results and Analytics**

**Actor(s):** Instructor  
**Description:** Instructor checks analytics and results of public exams.  
**Precondition:** Exams have been conducted.

**Normal Flow:**

1. Instructor navigates to analytics section.
2. System displays exam performance data.

**Alternate Flow:**

* If data is unavailable, the system notifies the user.

**Postcondition:** Instructor views analytics.

**10. Access Quiz Results and Reports**

**Actor(s):** Student  
**Description:** Student checks performance and results after taking a quiz.  
**Precondition:** Quiz attempt is completed.

**Normal Flow:**

1. User navigates to results section.
2. System displays quiz results and reports.

**Alternate Flow:**

* If results are delayed, the system notifies the user.

**Postcondition:** User views quiz results.

**11. Request Statistical Reports**

**Actor(s):** Admin  
**Description:** Admin requests system analytics and reports.  
**Precondition:** Admin is logged in.

**Normal Flow:**

1. Admin navigates to reporting section.
2. System generates statistical reports.
3. System displays the reports.

**Alternate Flow:**

* If report generation fails, the system logs an error.

**Postcondition:** Admin views system analytics.

**12. Handle System Errors (Error Control System)**

**Actor(s):** System  
**Description:** System logs and manages errors during processes.  
**Precondition:** System encounters an error.

**Normal Flow:**

1. System detects an error.
2. System logs error details.
3. System notifies relevant stakeholders.

**Alternate Flow:**

* If error persists, the system escalates to technical support.

**Postcondition:** Errors are managed and logged.

**3.4 Non-Functional Requirements**

**3.4.1 Execution Qualities**

* **Safety**  
  The system shall reject quizzes that are inappropriate, harmful, or offensive. Errors that could confuse students or undermine academic integrity shall be avoided.
* **Security**  
  The system shall protect user data through safe authentication (e.g., secure password policies and optional two-factor authentication). Sensitive data, including user credentials, shall be encrypted during both storage and transmission. User activities shall be logged for security auditing.
* **Usability**  
  The system shall provide a user-friendly and intuitive interface, making it accessible for users with limited technical skills. Accessibility features shall be supported, and user manuals/help guides shall be provided.
* **Performance Efficiency**  
  The system shall provide a response time of less than five seconds for common operations. It shall support up to 250 concurrent users without performance degradation and ensure scalability to handle increased demand.
* **Reliability & Availability**  
  The system shall ensure at least 99% uptime and allow automatic recovery from failures. Data shall be backed up daily to prevent loss. The system shall be accessible 24/7 except during scheduled maintenance, with downtime not exceeding 5 hours per month. Real-time notifications shall be provided during outages.

**3.4.2 Evolution Qualities**

* **Testability**  
  The system shall support comprehensive testing, including unit, integration, and performance testing, to ensure correctness and reliability.
* **Maintainability**  
  The system shall follow a modular architecture with proper documentation to enable easy bug fixes, updates, and feature additions. Updates shall be deployable with minimal downtime.
* **Portability**  
  The system shall be deployable across Windows, Linux, and macOS platforms and support both desktop and mobile web browsers. It shall be easily transferable to cloud environments.
* **Scalability**  
  The system shall allow horizontal and vertical scaling to handle workload increases. Future expansions shall support up to 10,000 users and integration with cloud services.
* **Interoperability**  
  The system shall integrate with existing educational tools and platforms. It shall support standard data exchange formats (e.g., JSON, XML) and provide APIs for third-party integration.
* **Legal and Compliance**  
  The system shall comply with relevant data protection regulations (e.g., GDPR). Copyrighted content shall not be used without proper licensing, and institutional policies on academic integrity shall be respected.

**3.5 Constraints**

**a. Resource Constraints**

* **Financial Resources:** The development and operation of the Quiz AI system must stay within the allocated budget, covering expenses such as development, AI model training, testing, deployment, and user training.
* **Human Resources:** Availability of team members skilled in software development, AI/ML, database management, and UX design may impact the project timeline.
* **Time Constraints:** The project must meet milestones for development, testing, and final submission within the semester or allocated timeline.

**b. Technological Constraints**

* **Platform Compatibility:** The system must work across modern web browsers on desktops, laptops, and mobile devices for students, instructors, and admins.
* **Database Performance:** The system must handle multiple concurrent quiz attempts, content uploads, and result queries efficiently.
* **AI and API Dependence:** Features such as quiz generation and content classification rely on AI models and stable API integrations.
* **Security Measures:** The system must use encryption, secure authentication, and input validation to protect user data and prevent unauthorized access.

**c. External Constraints**

* **User Availability:** Successful adoption depends on students, instructors, and admins actively using the system and engaging with training materials.
* **Data Quality:** AI-based quiz generation requires accurate, well-formatted content to produce meaningful quizzes.
* **Network Reliability:** Features like quiz attempts, content uploads, and notifications depend on stable internet connectivity.
* **Legal Compliance:** The system must respect data privacy regulations (like GDPR) and institutional rules regarding academic integrity.

**d. Operational Constraints**

* **Authentication Protocols:** Secure login and session management are required to prevent unauthorized access.
* **Data Synchronization:** Quiz attempts, results, and content uploads must sync across devices in real time.
* **Error Handling:** The system must provide clear error messages for invalid input, failed uploads, or unsuccessful quiz generation.
* **Audit Trail Maintenance:** Logs of quiz creation, user attempts, and system errors must be maintained for review and quality assurance.

**3.6 Software Diagrams**

A. Usecase Diagram

- The Use Case Diagram provides a high-level view of the system's functionalities and interactions with external actors.

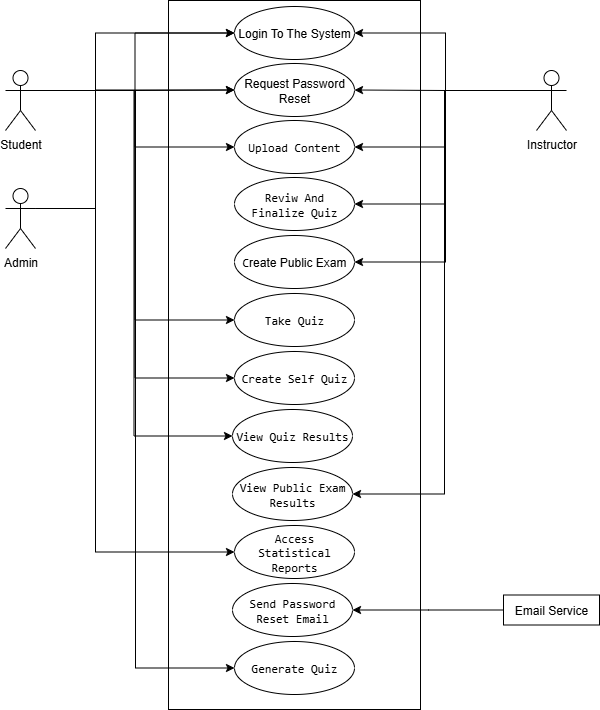


Figure 3.2: Use Case Diagram For Quiz AI System.

B. Context Diagram (Equivalent to DFD LV.0)

- The context diagram is an overview of an organizational system that shows the system boundaries, external entities that interact with the system, and the major information flows between the entities and the system.

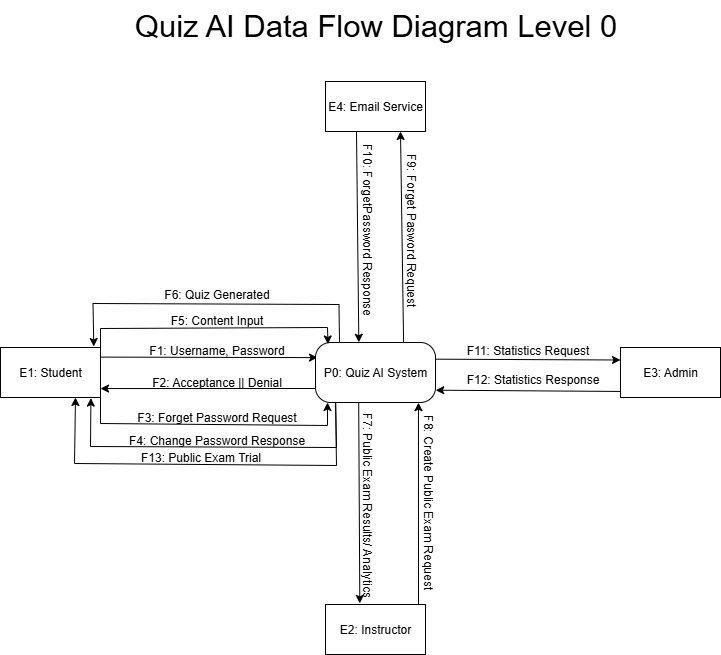


Figure 3.3: Context diagram of the proposed system.

Figure 3.3 presents the context diagram of the proposed Quiz AI System. This diagram illustrates the system's interactions with external entities, such as admin, instructors, and students. It provides a clear representation of data flow between the system and these actors, highlighting the system’s role in managing, and creating the quizzes by the system.

C. Data Flow Diagram Level 1

A Level 1 DFD shows the **detailed flow of data within the system**, breaking down major processes from Level 0 into more specific sub-processes. It illustrates **how data moves between actors, processes, and data stores**, helping readers understand the **internal workings of the system**. Level 1 DFDs provide clarity on **data handling, storage, and processing**, making it easier to follow each functional part of the system.

A diagram of a computer

AI-generated content may be incorrect.

**Figure 3.4: Level 1 DFD of the proposed system.**

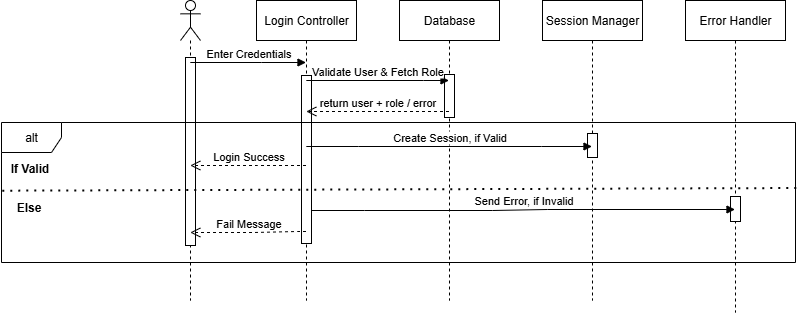
Figure 3.4 illustrates the Level 1 Data Flow Diagram (DFD) of the proposed Quiz AI System. This diagram depicts the main processes of the system, such as quiz creation, generating statistics, and error control. It shows how data flows between various processes, the database, and external entities, providing a clear understanding of the system’s detailed operations and its role in achieving quality assurance objectives at the Quiz AI System.

D. Sequence Diagrams:

A sequence diagram is a type of UML diagram that shows **how objects and actors interact over time** to complete a process. It illustrates the **order of messages**, **activations**, and **conditions** between actors (like users) and system components (like controllers or databases).

It helps readers understand **step-by-step workflows**, such as login, content upload, quiz generation, or exam attempts, by showing **who does what and in what order**.

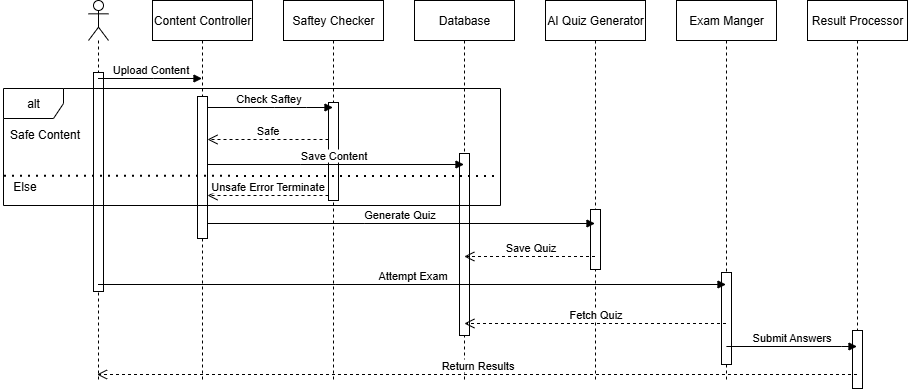
* + Login Sequence Diagram:



**Figure 3.5: Login Sequence Diagram.**

The login sequence diagram illustrates the step-by-step process a user follows to access the system. It shows how the **user interacts with the Login Controller**, how the system **validates credentials and fetches the user’s role from the database**, and how it **creates a session or returns an error**. Conditional fragments indicate the alternative flows for **valid and invalid logins.**

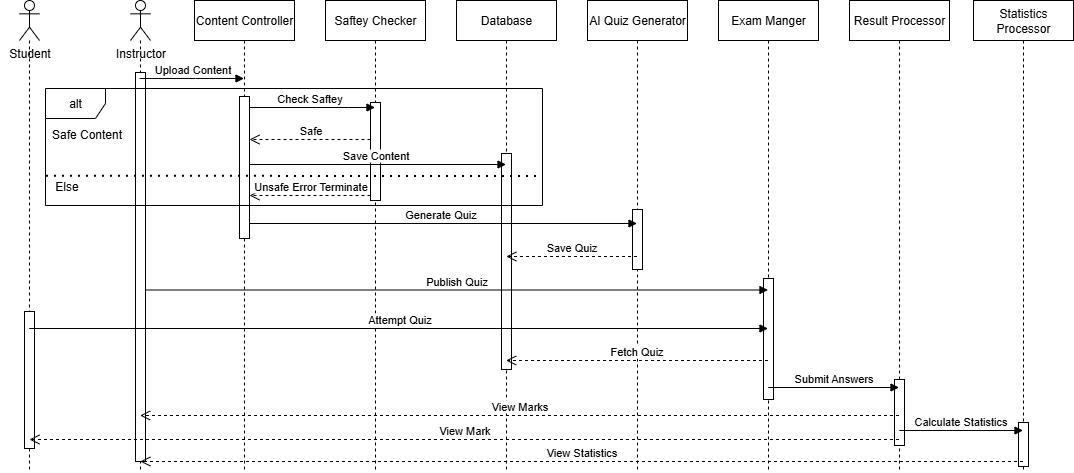
* + Student Attempt Exam Sequence Diagram:



**Figure 3.6: Student Attempt Exam Sequence Diagram.**

This sequence diagram shows how a student interacts with the system to **upload content, generate a quiz, attempt an exam, and receive results**. It illustrates the flow between the **student, content controller, safety checker, database, AI quiz generator, exam manager, and result processor**. Conditional fragments represent **alternative flows**, such as unsafe content or late exam submissions. The diagram provides a clear view of how the system manages content, generates quizzes, handles exam attempts, and returns scores and feedback to the student.

* + Instructor Quiz Management Sequence Diagram



**Figure 3.7: Instructor Quiz Management Sequence Diagram.**

This sequence diagram illustrates how an instructor interacts with the system to **upload content, ensure its safety, generate quizzes via AI, publish them, and monitor student performance**. It shows the flow between the **instructor, content controller, safety checker, database, AI quiz generator, exam manager, result processor, and stats processor**. Conditional fragments capture alternative flows, such as **unsafe content or unapproved quizzes**. The diagram clearly depicts how the instructor manages quiz creation, publication, and accesses **marks and statistical reports** for evaluation.