



**WEB BASED LEARNING ASSESSMENT SYSTEM**

**BY**

**MUDATHIR MUKTHAR MUHAMMAD**

**NAS/STE/20/1054**

**A PROJECT SUBMITTED TO THE DEPARTMENT  
OF  
SOFTWARE ENGINEERING AND CYBER SECURITY**

**COLLEGE OF COMPUTING AND INFORMATION SCIENCE,  
AL-QALAM UNIVERSITY KATSINA**

**IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD  
OF BACHELOR'S DEGREE  
IN  
SOFTWARE ENGINEERING**

**MAY, 2024**

## **CHAPTER THREE**

### **SYSTEM ANALYSIS AND DESIGN**

#### **3.0 Introduction**

This chapter provides a detailed analysis and design of the Learning Assessment System. It covers the software development model used, the requirement engineering processes, and the system design. Each section elaborates on the methodologies and tools employed to ensure the development of an efficient and user-friendly online testing platform.

In this chapter, all the requirements necessary for the software development process, namely functional and non-functional requirements, system architecture, use case diagram, activity diagram, class diagram, and entity relationship diagram (ERD).

#### **3.1 Software Development Model**

The Agile model was chosen for the development of the Learning Assessment System. Agile is an iterative and incremental approach to software development that emphasizes flexibility. The process begins with an initial planning phase, followed by multiple iterations or sprints, each resulting in a potentially shippable product increment.

Agile's iterative nature means that, I can quickly adapt to new insights or changes in requirements. For example, if I discover that a particular feature could be enhanced for better user experience, I can incorporate those changes in the next sprint. This approach ensures that the final product aligns closely with user expectations and requirements, even if those evolve during the development process.

Each sprint involves planning, designing, coding, testing, and reviewing, allowing for continuous improvement and adaptation to changing requirements. The justification for using the Agile model is its ability to accommodate evolving user needs and incorporate feedback quickly, ensuring that the final product aligns closely with user expectations and requirements. The iterative nature of Agile also helps in identifying and addressing issues early in the development process, leading to a more robust and reliable system. Using the Agile model in this solo project enables a structured yet flexible approach to development. It allows for the accommodation of evolving user needs, continuous improvement, and early

detection and resolution of issues. Ultimately, this approach ensures that the Learning Assessment System is developed efficiently and effectively, meeting its objectives and providing a valuable tool for online testing and assessment.

### **3.1.1 Benefits of Using Agile**

- 1. Customer-Centric Development:** Agile's ensures that the Learning Assessment System is developed with direct input from users and stakeholders, leading to a product that meets their needs more accurately.
- 2. Flexibility and Adaptability:** Agile allows for changes in requirements even late in the development process. This is particularly important for the Learning Assessment System, as educational requirements and technologies can evolve rapidly.
- 3. Faster Time-to-Market:** By delivering product increments at the end of each sprint, Agile enables the Learning Assessment System to be released to users sooner, allowing for early feedback and continuous improvement.
- 4. Improved Collaboration:** Agile fosters a collaborative environment where all team members and stakeholders are engaged in the development process, leading to better communication and teamwork.
- 5. Risk Mitigation:** The iterative nature of Agile helps in identifying and addressing risks early in the development process. Regular reviews and retrospectives ensure that potential issues are resolved promptly.
- 6. Quality Assurance:** Continuous integration and testing during each sprint help maintain a high level of quality in the system. Issues are detected and resolved quickly, resulting in a more stable and reliable product.

By adopting the Agile model, the development of the Learning Assessment System is more responsive to user needs, able to adapt to changing requirements, and capable of delivering a high-quality product in a timely manner. This approach ensures that the Learning Assessment System remains relevant and effective in providing online learning solutions for educational institutions.

### **3.2 Requirement Engineering**

Requirement engineering is the process of defining, documenting, and maintaining the requirements for a software system. It involves several key

activities, including requirements elicitation, analysis, specification, validation, and management.

Requirements Engineering ensures that the problem a client wants solved is clearly defined and the solution is both accurate and effective. Essentially, Requirements Engineering transforms a real-world problem into a clear specification for a highly functional solution.

### **3.2.1 Process of Requirement Engineering**

Requirements Engineering (RE) determines the requirements of software according to the needs of customers. Requirements engineering process includes:

- 1. Requirements Elicitation:** Requirements elicitation involves gathering requirements from stakeholders through various techniques such as interviews, surveys, and observations. For the Learning Assessment System, requirements were collected from potential users (students and administrators) and educational institutions.
- 2. Requirements Analysis:** This step involves analyzing the gathered requirements to ensure they are complete, consistent, and feasible. For the Learning Assessment System, this included identifying key functionalities such as user login, test-taking, score viewing, retaking tests, and test management by administrators.
- 3. Requirements Specification:** The analyzed requirements are documented in a clear and detailed manner. For the Learning Assessment System, a Software Requirements Specification was created, outlining all functional and non-functional requirements.
- 4. Requirements Validation:** This involves reviewing the requirements with stakeholders to ensure they accurately reflect their needs and expectations. For the Learning Assessment System, validation sessions were conducted with representatives from the user groups to confirm the requirements.
- 5. Requirements Management:** This ongoing process involves tracking and managing changes to the requirements throughout the project lifecycle. For the Learning Assessment System, a requirements management tool was used to document and track any changes or updates to the requirements.

### 3.3 Requirement Specification

A software requirements specification (SRS) is an extensive description of the planned software to be designed. It describes in detail what the proposed software will render and how it is supposed to perform. The SRS serves as a blueprint for both the development team and stakeholders, ensuring that all parties have a clear understanding of the system's requirements and expectations. It plays a crucial role in guiding the design, development, and testing phases of the software lifecycle.

The SRS of the Learning Assessment System is divided into two main categories: functional requirements and non-functional requirements.

#### 3.3.1 Functional Requirements

A functional requirement describes the functionality of a software and its components. Functionality is defined as a set of software behavior, inputs, and output. This section consists of the functional requirements for the Learning Assessment System.

1. **User Management:** The system allows for the management of user accounts by administrators.
2. **Test Management:** Administrators can add, remove, and manage tests.
3. **User Interaction:** Users can log in, take tests, view scores, and retake tests if necessary.
4. **Feedback Mechanism:** Users can provide feedback, and administrators can review this feedback.

#### 3.3.2 Non-Functional Requirements

Non-functional requirements specify the criteria that can be used to judge the operation of a software system, rather than specific behaviors. These requirements ensure the software is reliable, efficient, and maintainable. For the Learning Assessment System, the non-functional requirements include:

1. **Performance:** The system should provide a quick response to user actions.
2. **Scalability:** The system should be able to handle increasing volumes of data and users.
3. **Reliability:** The system should perform consistently over time.

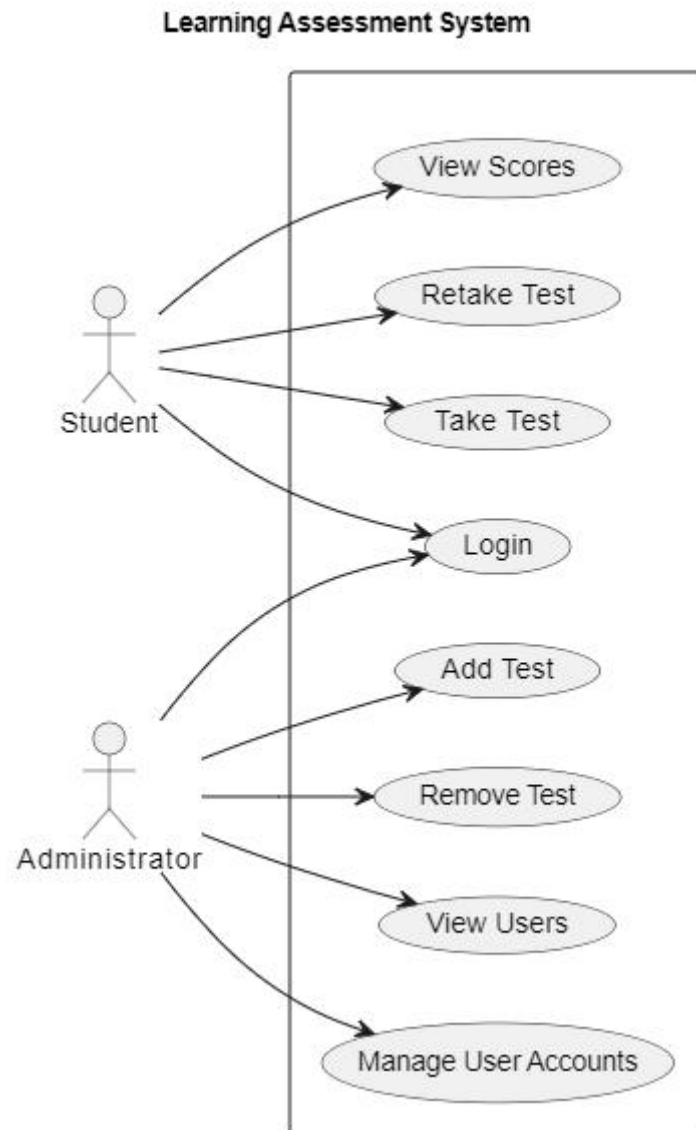
4. **Availability:** The system should be available 24/7 with minimal downtime.
5. **Recoverability:** The system should be able to recover quickly in the event of a failure.

### **3.4 System Design**

The system design involves creating models and diagrams to represent the system's architecture, data flow, and relationships. This phase focuses on creating a structured framework that defines the architecture, components, interfaces, and data necessary for the Learning Assessment System. The goal is to ensure the system is robust, scalable, and maintainable. Key design tools and methodologies used include Use Case diagrams, Data Flow Diagrams (DFD), and Entity Relationship Diagrams (ERD).

#### **3.4.1 Use Case Diagram**

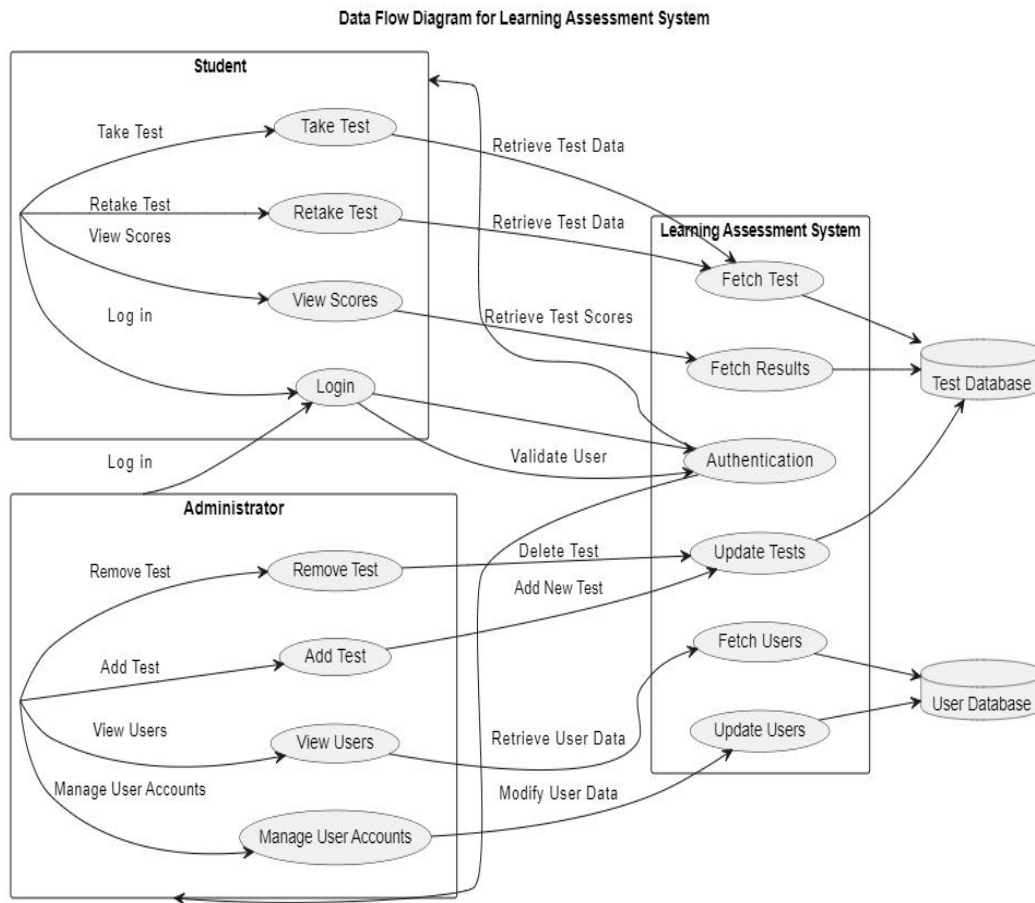
A use case diagram depicts a unit of functionality of a system. The essential goal of the use case diagram is to aid development teams in visualizing the functional requirements of the software, including the relationship of actors (e.g., students, administrators) to primary processes, as well as the relationships between different use cases. In this project, use case diagrams were employed during the requirements gathering phase to graphically represent the system's functional requirements.



**Figure 3.1:** Use Case Diagram

### 3.4.2 Data Flow Diagram

Data Flow Diagram (DFD) is a visual representation of the information flow through a process or system. DFDs help you better understand process or system operations to discover potential problems, improve efficiency, and develop better processes. Creating a Data Flow Diagram (DFD) involves illustrating the flow of data within your Learning Assessment System.



**Figure 3.2: Data Flow Diagram**

#### **Data Flow:**

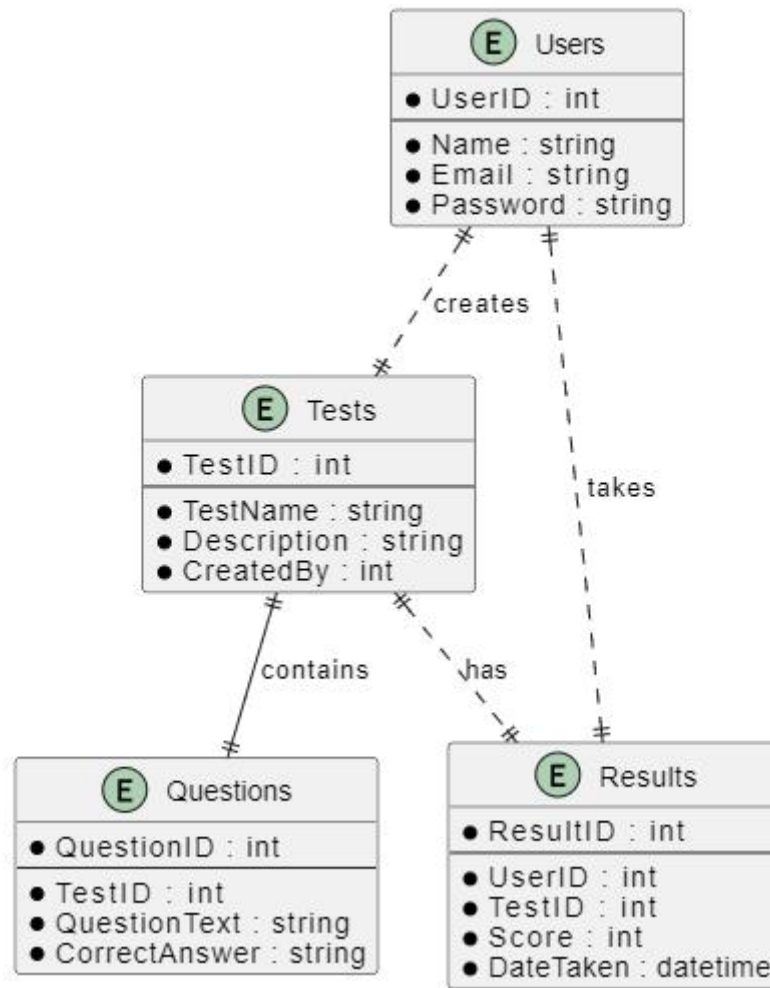
- Student and Administrator are represented as separate rectangles containing their interactions with the system.
- Learning Assessment System contains the processes within the system.
- Test Database and User Database are represented as databases where the system stores and retrieves data.
- Arrows indicate the direction of data flow between processes and data stores.

#### **3.4.3 Entity Relationship Diagram**

An entity relationship diagram (ERD) is a graphical representation of database tables and their relationships to each other, usually used in software development projects regarding the organization of data within the databases or information systems. In the context of an Learning Assessment System., an ERD helps illustrate the relationships between key entities such as Students, and Administrators.



### Entity Relationship Diagram for Learning Assessment System



**Figure 3.3:** Entity Relationship Diagram

#### ERD Explanation:

- **Users:** This entity represents the users of the system. Each user has a unique ID, name, email, and password.
- **Tests:** This entity represents the tests in the system. Each test has a unique ID, name, description, and the ID of the user who created it.
- **Questions:** This entity represents the questions associated with each test. Each question has a unique ID, the ID of the test it belongs to, the question text, and the correct answer.
- **Results:** This entity represents the results of tests taken by users. Each result has a unique ID, the ID of the user who took the test, the ID of the test, the score, and the date the test was taken.

#### Relationships:

- A User can take multiple Results.

- A Test can contain multiple Questions.
- A User can create multiple Tests.
- A Test can have multiple Results.

### **3.5 Chapter Summary**

This Chapter focuses on the analysis and design of the Learning Assessment System. It begins by outlining the adoption of the Agile software development model for its iterative and flexible approach, ensuring adaptability to evolving requirements and close alignment with user expectations through continuous feedback. The chapter then dives into requirement engineering, emphasizing activities such as elicitation, analysis, specification, validation, and management to capture and define stakeholder needs effectively.

The requirement specification section outlines functional requirements (user management, test management, user interactions, feedback mechanisms) and non-functional requirements (performance, scalability, reliability, availability, recoverability). System design methodologies, including Use Case diagrams, Data Flow Diagrams (DFD), and Entity Relationship Diagrams (ERD), are employed to visualize system architecture, data flow, and database relationships.

Overall, Chapter Three provides a structured framework for developing an efficient and user-friendly online testing platform, ensuring that the Learning Assessment System meets its objectives through thorough analysis.