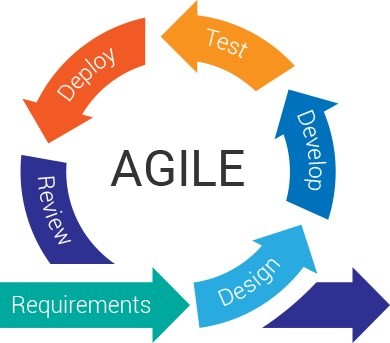
**Software Development Life Cycle Model**

The Agile methodology depicted in Figure 1 as an iterative approach to software development that emphasizes flexibility, collaboration, and customer satisfaction.

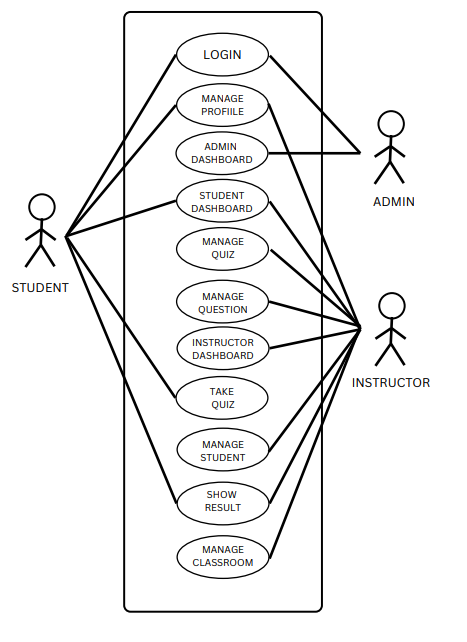
****

***Figure 1****: Software Development Life Cycle*

**Requirements:** The researchers requested permission to Mr. Stephen Duane Varela a BSCS instructor at North Negros College to conduct the study during the Requirements Gathering phase. We provided the scope and suggested potential solutions.

**Design:** Based on the requirements received, the design process included developing system architecture and design standards. This phase focused on the software's structure and functionality. Create an architecture and flow chart using Canva to trace everything we do or the system's architecture. This section allows us to visualize our system's parts and what must be done. We also design our database and ERD (Entity Relationship Diagram) during this phase.

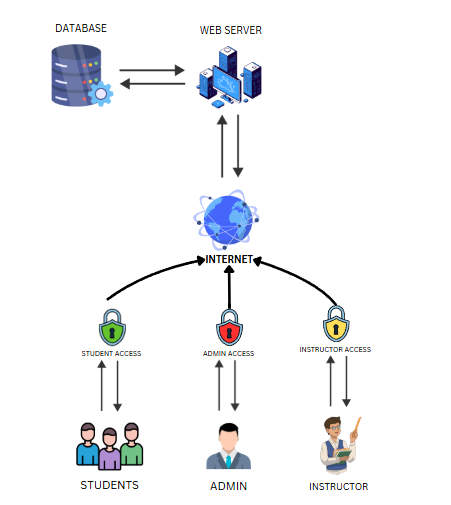
**User Design**

****

**Figure 2:** *Use Case Diagram*

The figure above illustrates how the user might interact with the system. It is a diagram representation of user, instructor and admin interactions with the system.

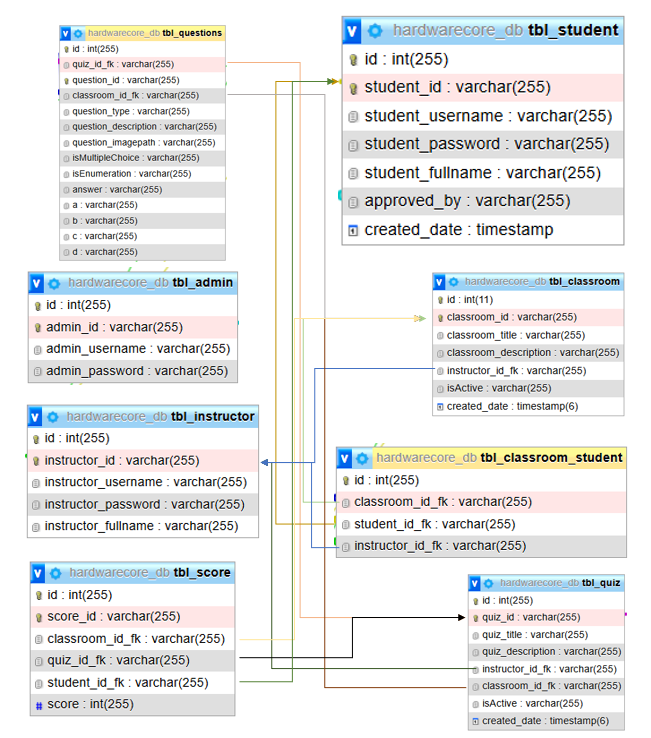
**System Architecture Design (Physical Design)**

****

***Figure 3****: System Architecture Design HardwareCore:*

*An Exclusive Quiz and Reviewer System for Computer System Servicing Students at North Negros College*

**Entity Relationship Diagram**

**

***Figure 3****: Entity Relationship Diagram HardwareCore:*

*An Exclusive Quiz and Reviewer System for Computer System Servicing Students at North Negros College*

**Development**

The development of HardwareCore: An Exclusive Quiz and Reviewer System for Computer System Servicing Students at North Negros College was carried out using PHP, jQuery, JavaScript, Bootstrap, and HTML. This phase focused on implementing core backend functionalities, database management, and frontend interfaces to ensure seamless user interaction.

**Software and Hardware Requirements**

|  |  |
| --- | --- |
| **Software Requirements** | **Use** |
| Web Brower | Access the system's web interface. |
| Windows OS | Make use of Windows 11. |
| Visual Studio Code | Coding-related integrated development environment (IDE). |
| Hypertext Markup Language | In order to organize the online interface. |
| Cascading Style Sheet  Database (MYSQLI)  Bootstrap  JQuery | Use for Front end styling and design.  To Manipulate system data  Use speed up styling  Use to Front End |
| PHP | Use in backend development. |
| JavaScript | Use for client-side scripting. |
|  |  |

**Table 1:** Software Requirements

**Hardware requirement**

The table below shows the hardware requirements, which include a 256 GB SSD, a processor, and RAM, to ensure the smooth operation of the applications required for system creation and operation.

|  |  |
| --- | --- |
| **Hardware Requirements** | **Specification** |
| Storage | 512 GB |
| Processor | Intel i5 11th Gen |
| RAM | 16 GB |
|  |  |
|  |  |

**Table 2:** Hardware Requirements

The system architecture was designed with a frontend built using HTML, CSS, Bootstrap, and JavaScript to provide an interactive and user-friendly experience. The backend was developed using PHP, handling server-side logic, data processing, and authentication. A MySQL database was implemented to manage user data, classrooms, quizzes, and student responses efficiently.

Key Features Developed:

1. User Role Management:
   * Admin: Manages instructor registrations and system users.
   * Instructor: Creates and manages classrooms, quizzes, and student records.
   * Student: Registers, joins classrooms, takes quizzes, and reviews materials.
2. Classroom Management:
   * Instructors can create and manage multiple classrooms, ensuring students are properly sorted into their respective sections.
   * Students can view their enrolled classrooms, track activities, and access quizzes and reviewer materials.
3. Dynamic Quiz System:
   * Instructors can create customizable quizzes with identification and multiple-choice questions.
   * Flexible quiz settings allow instructors to define the number of enumerations, identification, and multiple-choice questions per test.
4. Exam Expiration & Timed Quizzes:
   * A time-based expiration system was implemented to encourage students to complete quizzes within the designated period.
   * Once the expiration time is reached, the quiz becomes inaccessible.
5. Quiz & Classroom Code Sharing:
   * Each classroom and quiz is assigned a unique code that students can enter to gain access.
   * This feature simplifies enrollment and quiz participation.
6. Student Performance Tracking & Reports:
   * After completing a quiz, students can instantly view their scores.
   * Instructors can generate reports based on student quiz performance.
7. Data Storage & Security:
   * Secure SQL database management ensures proper storage and retrieval of user and quiz data.
   * User authentication and authorization mechanisms were implemented to prevent unauthorized access.

This development phase ensured that all core functionalities were built and optimized for a responsive, interactive, and efficient learning experience. The system was designed to enhance student engagement while providing instructors with powerful tools to manage their classrooms effectively.