**Problem 1: Library Management System**

1. Class Diagram:

**A screenshot of a computer

Description automatically generated**

2. Database Diagram:

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

The tables ‘Book’, ‘Patron’, and ‘Borrowing’ are linked using foreign keys (‘book\_id’ and ‘patron\_id’) in the ‘Borrrowing’ table, establishing relationships between books, patrons, and borrowings. The primary keys are: ‘bookID’ for the ‘Book’ table, ‘patronID’ for the ‘Patron’ table and ‘borrowID’ for the ‘Borrowing’ table.

This schema provides a foundation for the Library Management System’s database, linking books to patrons through borrowings and allowing for efficient management of book transactions and patron details.

**Problem 2: Online Quiz System**

1. **Logical Design:**

Question Organization:

1. **Question Structure**: Each question should contain:

* Question number or unique ID
* Question text
* Multiple-choice options (4)
* Correct answer

1. **Question Pool:** Store the 50 questions in an array or object to easily access and track them during the quiz.

A computer screen shot of a code

Description automatically generated

User Progress:

1. **User Data:** Store user-related data such as:

* User’s selected answers
* Current question number
* User’s score

1. **Handling Progress**:

* When a user selects an answer, update the user’s progress by storing their choice and moving to the next question.
* Calculate and update the score based on correct answers.

**2. Algorithm Implementation (Partial Code):**

**A computer screen shot of text

Description automatically generated**

**A computer screen with colorful text

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A computer screen with text

Description automatically generated**

**A screen shot of a computer program

Description automatically generated**

**A screen shot of a computer code

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a quiz results

Description automatically generated**

Click ‘Try Again’:

A screenshot of a computer

Description automatically generated

Click ‘Go To Home’:

A screenshot of a computer

Description automatically generated

**3. Class and Database Representation (Explanation Only):**

Class Diagram:

1. **Question Class:**

* Attributes: question number or ID, question text, options, answer

1. **User Class:**

* Attributes: selected answers (attempt), score (correct answers/total questions), correct answers, wrong answers, progress (percentage), maybe a user ID

OR

1. **Question Class:**

* Attributes:
* questionID(or number): int
* questionText: string
* option: string[4]
* correctAnswerIndex: int

1. **Quiz Class:**

* Attributes:
* questionPool: Question []
* currentQuestionIndex: int
* userScore: int
* userAnswers: {questionID: int, userChoice: int} []

1. **QuizInterface (UI) class:** 
   * Responsibilities:

* Displaying questions, options, and user interface elements
* Handling user input and interactions

1. **QuizController** 
   * Responsibilities:

* Managing the flow of the quiz
* Controlling the logic for question retrieval, score calculation, and user progress

Relationships:

* + Quiz-Question (Composition): The quiz class contains a collection of Question instances representing the pool of questions available for the quiz.
  + Quiz-QuizInterface(Association): The quiz class interacts with the QuizInterface class to display questions and handle user interactions.
  + Quiz-QuizController(Association): The quiz class utilizes the QuizController to control the flow and logic of the quiz process.

Database Schema:

1. **Tables:**

* Question tables with columns for ID, Question Text, Options, Correct Answer
* Users table with columns for User ID, Selected Answers, Score, Progress

1. **Relationships:**

* One-to-Many relationship between Users and Selected Answers (each user can have multiple selected answers)
* No direct relationship between Questions and Users (questions are common for all users)

Flow of Data:

1. **During quiz:**

* User interacts with the web interface.
* User’s selected answers and progress are stored and updated.

1. **After quiz:**

* User’s final score and answers can be stored in the database for future reference or analysis.