Learning Management System Design Document

2/26/2023

Version 1.0

Table of Contents

Table of Contents **2**

System Overview **3**

References 3

Environment Overview **4**

User Interface **4**

Data Storage **17**

Class Diagram – Structural Design **18**

Sequence Diagrams - Dynamic Model **19**

Scenario 1: Create a New Course 19

Scenario 2: Taking a Module 20

Scenario 3: Create a Comment 21

System Overview

Our LMS system targets users who seek proficiency certification for computer programming through the use of teacher-generated courses. Our system is accessible to all users at no cost and is designed for use both inside the classroom as well as by a broader audience interested in learning coding and its applications.

Courses will be created by teachers, and stored in a course list by their name. Users will have access to all teachers courses and have the ability to search for courses by teacher name. Administrators will oversee function and user input to the system.

References

- LMS Requirements Document

Environment Overview

This system will run on one computer, users will have access through the internet. Our program will compile and run on java. The program will include but not be limited to a terminal and java compiler, including a console and IDE not unlike visual studio code.

User Interface

This application is a Java Console application. This is a text-based application handled through the terminal. In these examples, input that would be user/case specific will be indicated in brackets (“[ ]”) with the description of what that text would be in them. A zip file of the UI example text files beyond the scenarios listed have been added as an additional attachment.

Scenario 1: Creating an account and Logging in

```
---- Coding Learning Management System ----

Welcome! What would you like to do?

1. Login
2. Register
0. Exit
```

Option 2 is selected:

```
 ---- Register ----

Enter your email address:

[email]
```

```
  Enter your username:

  [username]
```

```
 ---- Register ----

For security reasons, your password is required to:
> be at least 8 characters in length
> include at least 1 capital letter
> include at least 1 lowcase letter
> include at least 1 special character (Ex: !, @, #, $...)

Enter your password:

[password]
```

If the password does not meet the requirements:

```
your password does not meet the requirements. Please enter a new password.

[password]
```

If the password does meet the requirements:

```
---- Register ----

Success! [username] has been created. Type 0 to return to the welcome screen and log in.
```

```
---- Coding Learning Management System ----

Welcome! What would you like to do?

1. Login
2. Register
0. Exit
```

Option 1 is selected:

```
---- Login ----

Enter your username:

[username]
```

```
  Enter your password:

  [password]
```

If the credentials are invalid:

```
---- Login ----

Oops! The username and password do not match.

Enter your username:
```

```
  Enter your password:

  [password]
```

If the credentials are valid:
Student Account:

```
---- Coding Learning Management System ----

Welcome, [username]! What would you like to do?

1. Go to Course List
2. Search from course database
3. Logout
0. Exit
```

Teacher account:

```
---- Coding Learning Management System ----

Welcome, [username]! What would you like to do?

1. Go to Course List
2. Create a new course
3. Search from course database
4. Logout
0. Exit
```

Admin Account:

```
---- Coding Learning Management System ----

Welcome, [username]! What would you like to do?

1. Search from course database
2. Logout
0. Exit
```

Scenario 2: Creating a Course

```
---- Course Creation ----

Enter the name of your course

[course name]

Enter a course description

[Course description]

Enter the difficulty of your course

1. Beginner
2. Intermidiate
3. Experienced

[1]

Course has been created
```

```
    ---- [Course name] ----

    | [Course Author] |

    [Course description]

    This course has 0 modules

    What would you like to do?

    1. Add a module
    2. Create end-of-course quiz
    0. Back
```

Option 1 is selected

```
Enter the name of your module

[module name]

Enter the module description

[module description]
```

```
 | End-of-Module Quiz Creation |

 The quiz currently has 0 questions

 1. Add a question
 2. Complete Quiz Creation

 1
     Enter the question

     [Question]

     Enter answer choice

     [answer choice]
```

```
 [question]

 The answer choices are:
 1. [answer choice 1]

 There is no correct answer selected

 1. Add another answer choice
 2. Select correct answer
 3. Complete Question creation
```

If the user selects option 3 without more than one answer choice

```
 You must have at least 2 answer choices before completing question creation

 [question]

 The answer choices are:
 1. [answer choice 1]

 There is no correct answer selected

 1. Add another answer choice
 2. Select correct answer
 3. Complete Question creation
```

Option 1 is selected

```
Enter answer choice

[answer choice]
```

```
[question]

The answer choices are:
1. [answer choice 1]
2. [Answer choice 2]

There is no correct answer selected

1. Add another answer choice
2. Select correct answer
3. Complete Question creation
```

If the user tries to select option 3 without selecting a correct answer

```
You must select a correct answer before completing question creation

[question]

The answer choices are:
1. [answer choice 1]
2. [answer choice 2]

There is no correct answer selected

1. Add another answer choice
2. Select correct answer
3. Complete Question creation
```

Option 3 is selected

```
[question]

The answer choices are:
1. [answer choice 1]
2. [answer choice 2]

There is no correct answer selected

1. Add another answer choice
2. Select correct answer
3. Complete Question creation

[2]

Which answer is the correct answer?

[1]
```

```
[question]

The answer choices are:
1. [answer choice 1]
2. [answer choice 2]

The correct answer is [answer choice 1]

1. Add another answer choice
2. Select correct answer
3. Complete Question creation
```

Option 3 is selected

```
Question has been added

The quiz currently has 1 question

1. [Question]

1. Add a question
2. Remove a question
3. Complete Quiz Creation
```

Option 3 is selected

```
module created

| [module name] |

[Module Description]

This module has 0 sections

What would you like to do?

1. Add a section
2. Edit end-of-module quiz
0. Back
```

Option 1 is selected

```
Enter the section name

[enter section name]

Enter the section content

[content]
```

```
| [module name] |

[Module Description]

This module has 1 section

[section name]

What would you like to do?

1. Add a section
2. Remove a section
3. Edit end-of-module quiz
0. Back
```

Option 0 is selected

```
---- [Course name] ----

| [Course Author] |

[Course description]

This course has 1 module

[Module name]

What would you like to do?

1. Add a module
2. Remove a module
3. Create End-of-Course Quiz
0. Back
```

Option 3 is selected

```
| End-of-Module Quiz Creation |

The quiz currently has 0 questions

1. Add a question
2. Complete Quiz Creation
```

Option 1 is selected

```
Enter the question

[Question]

Enter answer choice

[answer choice]
```

```
| [question] |

The answer choices are:
1. [answer choice 1]

There is no correct answer selected

1. Add another answer choice
2. Select correct answer
3. Complete Question creation
```

Note: The same errors apply to the End-of-course Quiz as it does the End-of-module quiz
Option 1 is selected

```
Enter answer choice

[answer choice]
```

```
| [question] |

 The answer choices are:
 1. [answer choice 1]
 2. [Answer choice 2]

 There is no correct answer selected

 1. Add another answer choice
 2. Select correct answer
 3. Complete Question creation
```

Option 2 is selected

```
  | [question] |

   The answer choices are:
   1. [answer choice 1]
   2. [answer choice 2]

   There is no correct answer selected

   1. Add another answer choice
   2. Select correct answer
   3. Complete Question creation

  [2]

   Which answer is the correct answer?

   [1]
```

```
| [question] |

The answer choices are:
1. [answer choice 1]
2. [answer choice 2]

The correct answer is [answer choice 1]

1. Add another answer choice
2. Select correct answer
3. Complete Question creation
```

Option 3 is selected

```
    Question has been added

    The quiz currently has 1 question

    1. [Question]

    1. Add a question
    2. Remove a question
    3. Complete Quiz Creation
```

Option 3 selected

```
   ---- [Course name] ----

   | [Course Author] |

   [Course description]

   This course has 1 module

   [Module name]
   End-of-Course Quiz

   What would you like to do?

   1. Add a module
   2. Remove a module
   3. Edit End-of-Course Quiz
   0. Back
```

Scenario 3: Searching for a course by Teacher

```
---- Course Search ----

What would you like to search by?

1. Search by Teacher
2. Search by Difficulty
3. Search by Course name
0. Back
```

Option 1 is selected

```
Enter the name of the Teacher

[Teacher name]
```

```
There are 2 courses by [Teacher name]

> [Course 1]
> [Course 2]

What would you like to do?

1. View course
2. Add course to course list
0. Back
```

Option 2 is selected

```
Which course would you like to add?

[course 1]

Course 1 has been added to your course list
```

If the course has already been added

```
Which course would you like to add?

[course 1]

[Course 1] is already in your course list. Select another course.
```

Scenario 4: Leaving a comment on a course

```
| Comments |

Comments: 1

1. [Commenter username]

   [comment]

   [Comment Replies]

What would you like to do?

1. Add a comment
2. Reply to a comment
0. Back
```

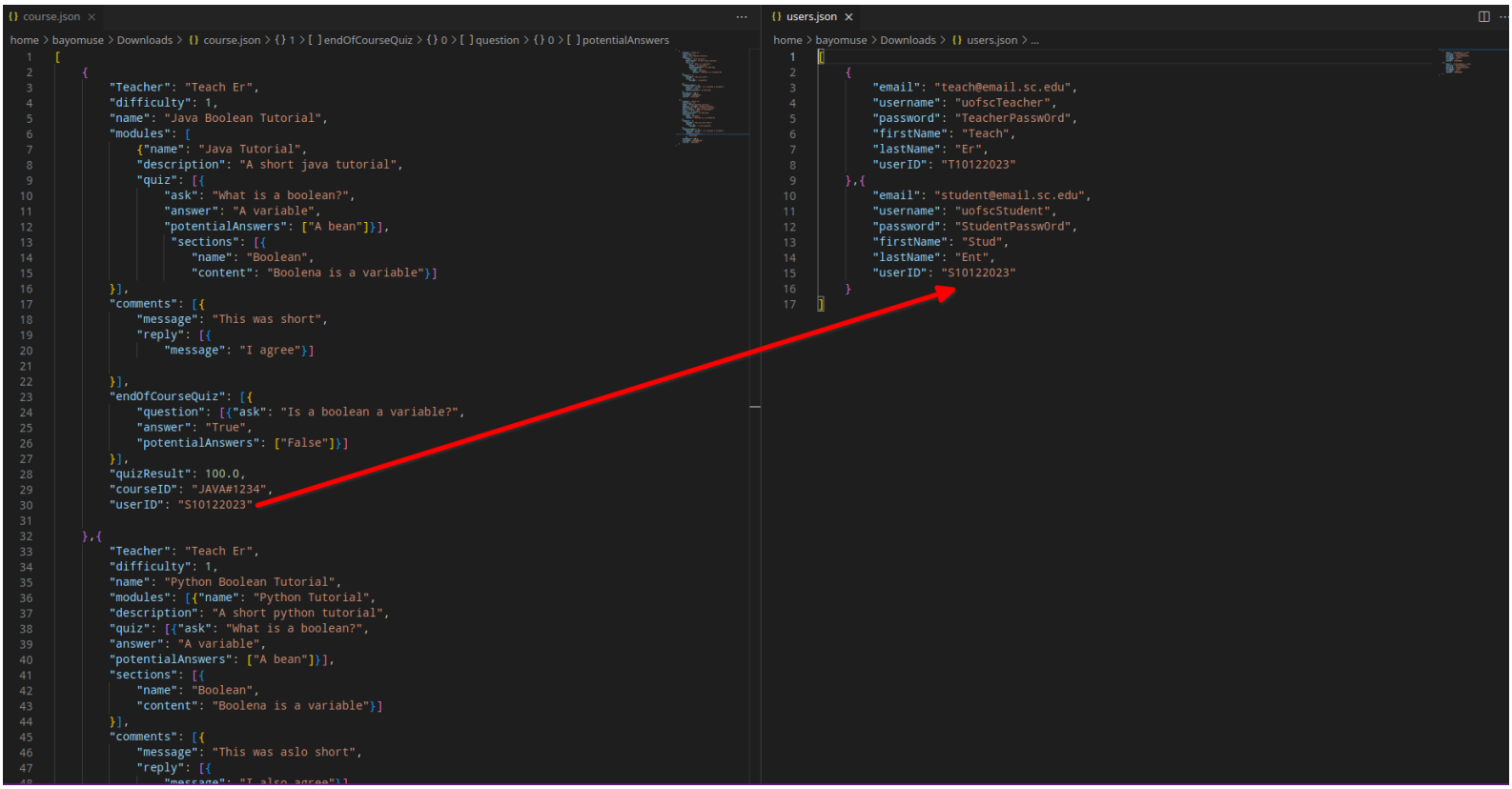
Option 1 is selected

```
Type in the contents of your comment

[comment content]

    comment added
```

```
   ---- Intro to JavaScript ----

   | Comments |

   Comments: 2

  1. [Commenter username]

     [comment]

     [Comment Replies]

  2. [Commenter username]

     [comment]

     [Comment Replies]

   What would you like to do?

   1. Add a comment
   2. Reply to a comment
   0. Back
```

Data Storage

The data for our system is stored in two JSON files: One for users, and one for courses. Each json file holds different attributes for the classes. The Course json holds multiple ArrayLists for objects such as modules and comments. Both Users and Courses have a unique identifier. Each course instance is connected by the user id of the user that is taking that course. The system will access the data through a UserList and CourseList class respectively, and will be able to save and delete data through a DataReader and a DataWriter class.



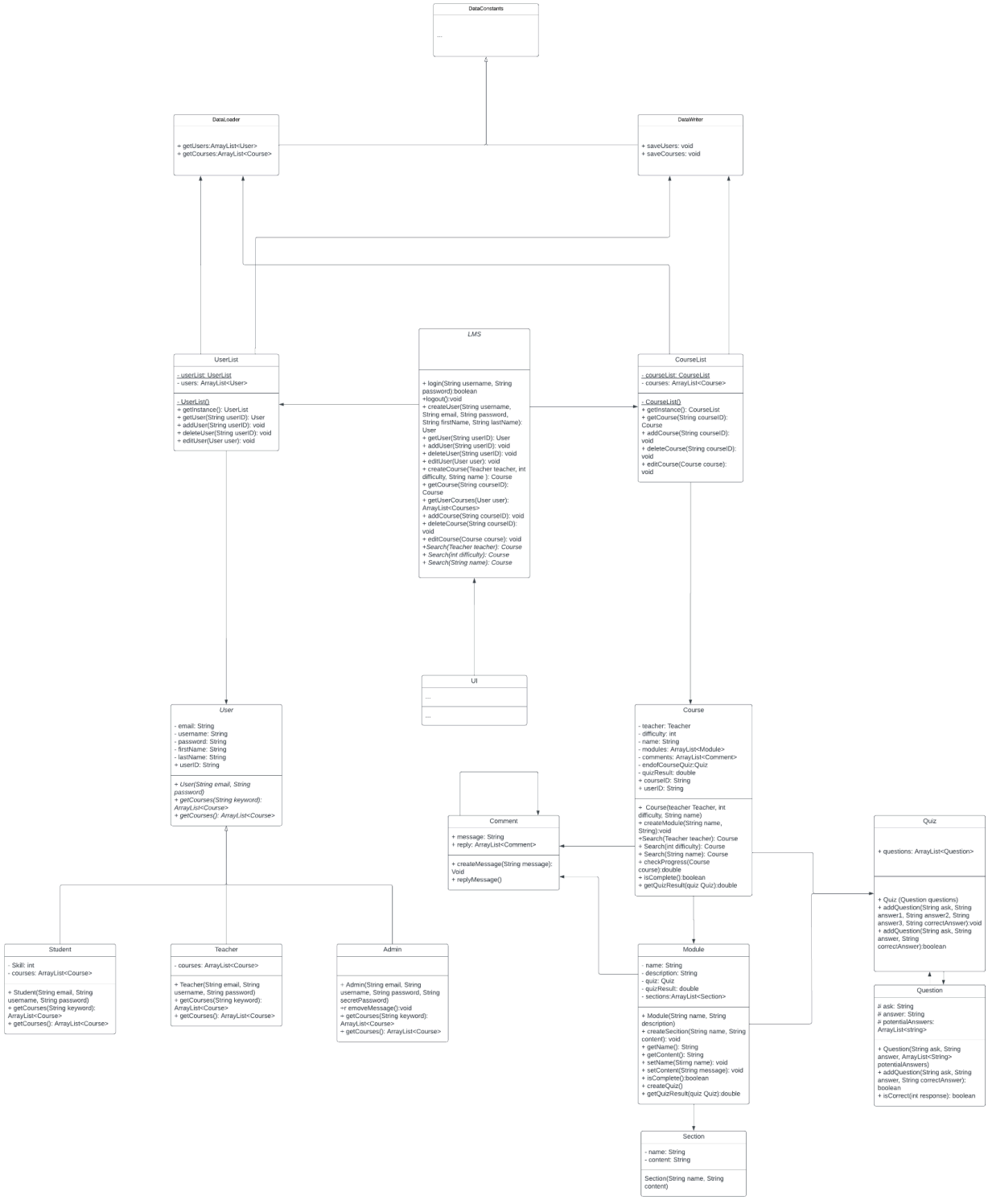
Users.json

```
[
    {
        "email": "teach@email.sc.edu",
        "username": "uofscTeacher",
        "password": "TeacherPassw0rd",
        "firstName": "Teach",
        "lastName": "Er",
        "userID": "T10122023"
    },{
        "email": "student@email.sc.edu",
        "username": "uofscStudent",
        "password": "StudentPassw0rd",
        "firstName": "Stud",
        "lastName": "Ent",
        "userID": "S10122023"
    }
]
```

Course.json


```
[
    {
        "Teacher": "Teach Er",
        "difficulty": 1,
        "name": "Java Boolean Tutorial",
        "modules": [
            {"name": "Java Tutorial",
            "description": "A short java tutorial",
            "quiz": [{
                "ask": "What is a boolean?",
                "answer": "A variable",
                "potentialAnswers": ["A bean"]}],
                "sections": [{
                   "name": "Boolean",
                   "content": "Boolena is a variable"}]
        }],
        "comments": [{
            "message": "This was short",
            "reply": [{
                "message": "I agree"}]

        }],
        "endOfCourseQuiz": [{
            "question": [{"ask": "Is a boolean a variable?",
            "answer": "True",
            "potentialAnswers": ["False"]}]
        }],
        "quizResult": 100.0,
        "courseID": "JAVA#1234",
        "userID": "S10122023"

    },{
        "Teacher": "Teach Er",
        "difficulty": 1,
        "name": "Python Boolean Tutorial",
        "modules": [{"name": "Python Tutorial",
        "description": "A short python tutorial",
        "quiz": [{"ask": "What is a boolean?",
        "answer": "A variable",
        "potentialAnswers": ["A bean"]}],
        "sections": [{
            "name": "Boolean",
            "content": "Boolena is a variable"}]
        }],
        "comments": [{
            "message": "This was aslo short",
            "reply": [{
                "message": "I also agree"}]
        }],
        "endOfCourseQuiz": [{
            "question": [{"ask": "Is a boolean a variable?",
            "answer": "True",
            "potentialAnswers": [
                "False"]}]
            }],
        "quizResult": 100.0,
        "courseID": "PYTHON#1234",
        "userID": "S10122023"
    }
]
```

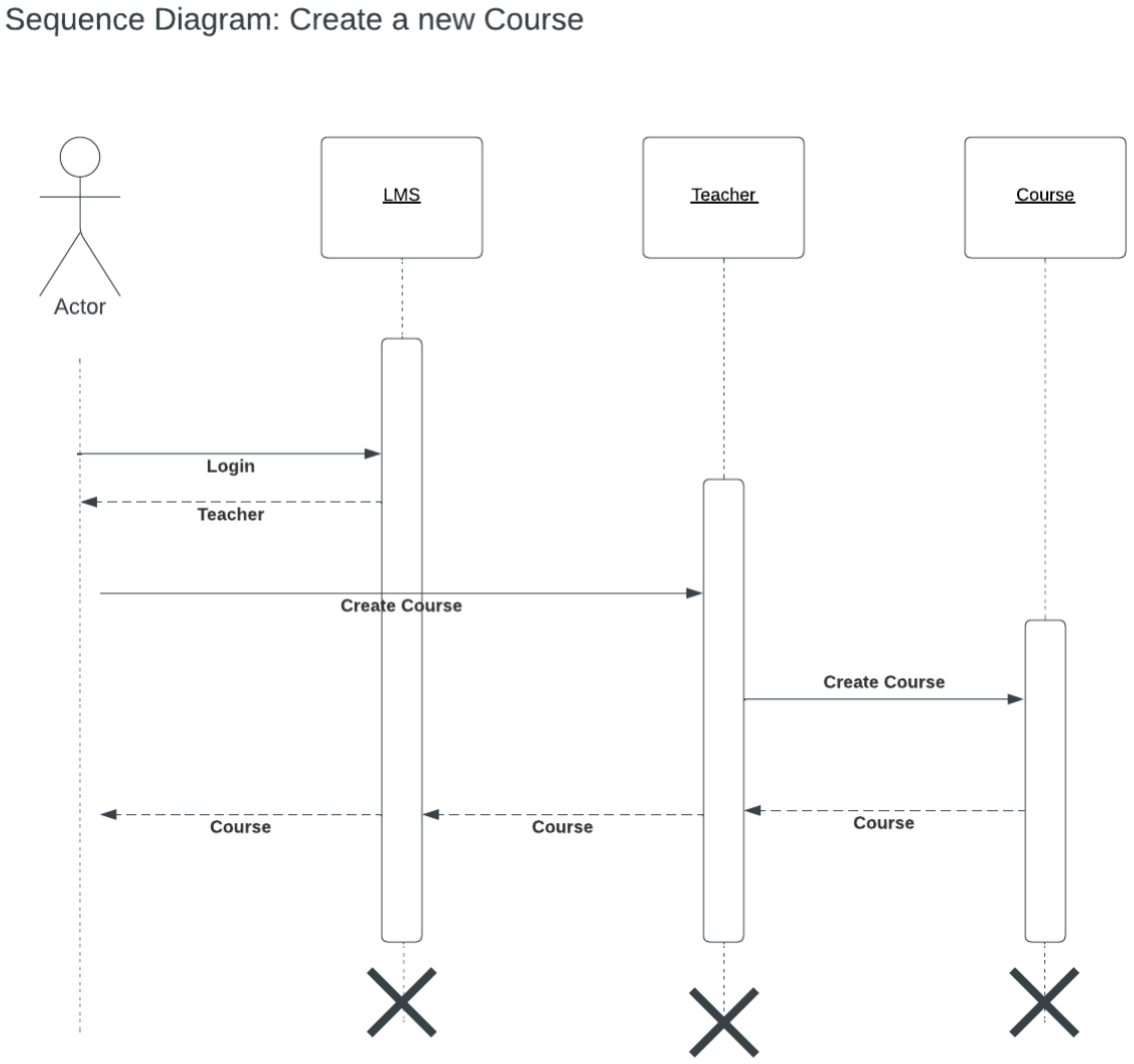
Class Diagram – Structural Design



Sequence Diagrams - Dynamic Model

Scenario 1: Create a New Course

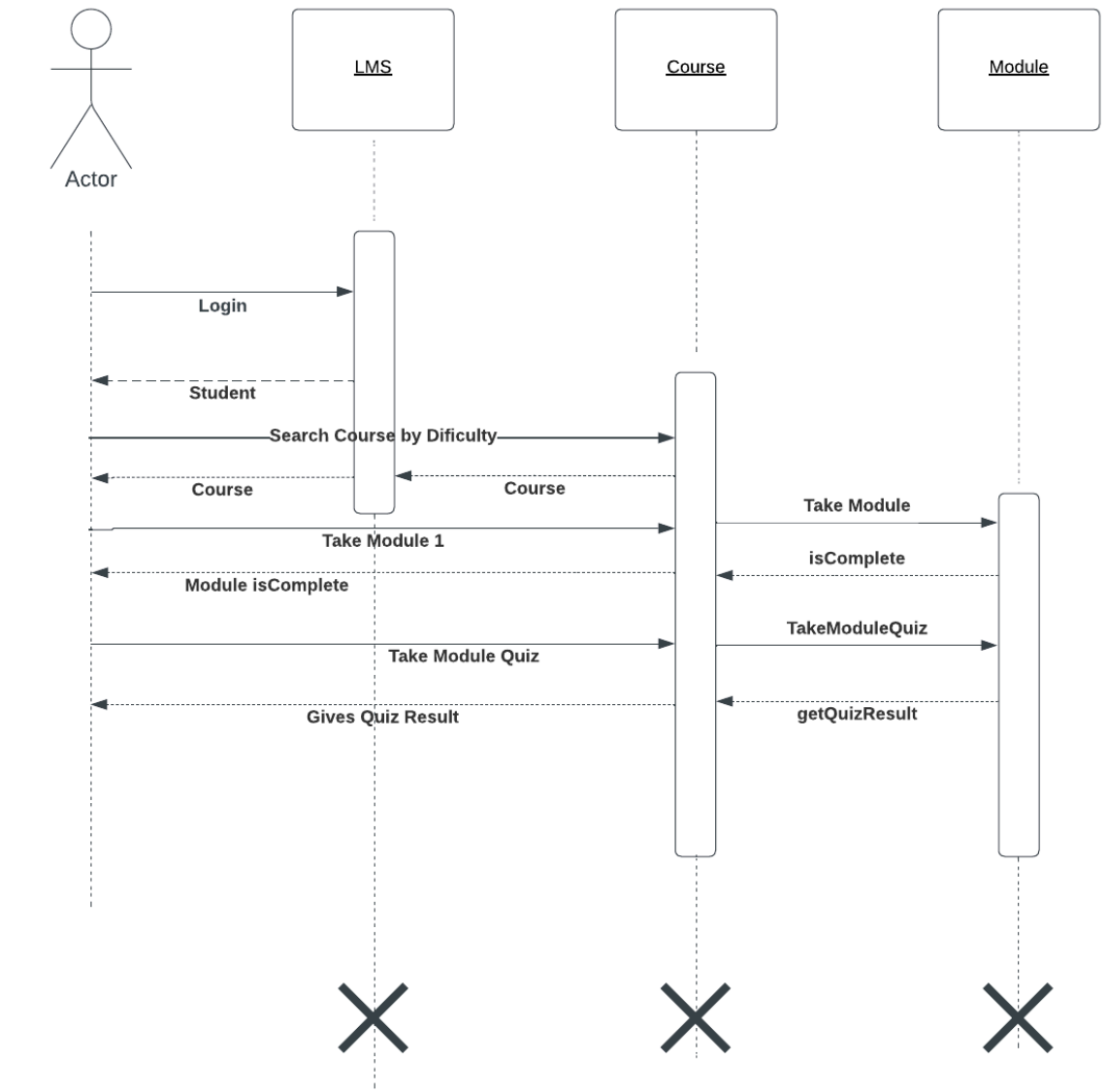
The user is a teacher who is creating a new course. They first log into their account. They then select that they want to create a course. The request goes through the Teacher, then to the Course, then finally the system. The Teacher will input each property of the course, including the name, difficulty, the modules, the sections, and the quizzes. The system will then create the course and store it in the JSON database.



Scenario 2: Taking a Module

The user logs in as a student. They search a course by the difficulty level and they system returns to them a list of courses that are that difficulty. After adding the course, the user goes into the course and selects a module to take. They then take the module and the system keeps track of their progress. After they have completed all of the sections for that module, they are able to take the end-of-module quiz. After they take the quiz, the system grades it and returns the result.

Sequence Diagram: Take a Module



Scenario 3: Create a Comment

The user logs in. All three types of accounts can create a comment. They search for a course by the teacher. The system returns a list of courses for the user to choose from. The user then picks a course they want to comment on and creates the message. After they submit the message, it is sent to the system and stored.

Sequence Diagram: Create a comment