

TeamUrameshi

02/26/2023

Code Cruncher Learning Management System Design Document

System Overview

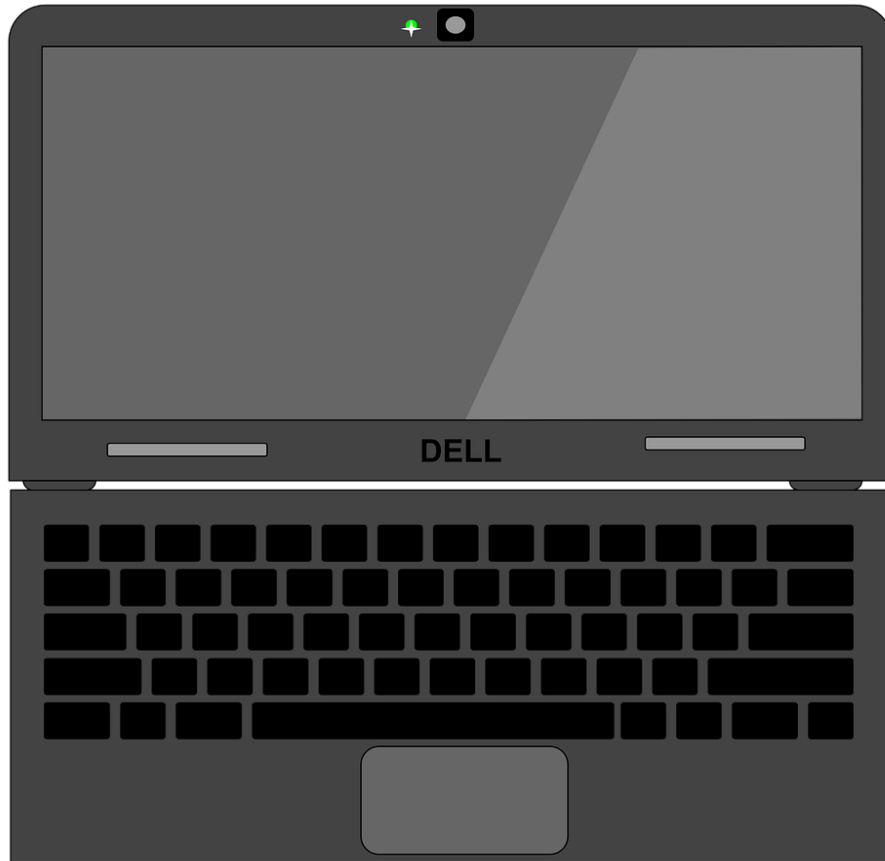
The purpose of our learning management system is to provide a remote programming education that teaches JavaScript, Python, and C# at any pace. Our learning management system will have modules for each respective programming language, and each module will have lessons, quizzes for those lessons, and a final exam at the end of the module. Our target users are students, teachers, and admins. The environment will be a text based program and users will be able to interact with the learning management system through the java console through a series of text prompts.

References

- Code Cruncher Requirement Document
- Code Cruncher System UML PDF

Environment Overview

The system will run on computer consoles/terminals and be purely text based. Only a keyboard shall be required to access and fully utilize the system. Such simple environment requirements will make our learning management system accessible to almost anyone in the world who wants to learn programming languages as long as they have access to a computer.



The system will be a collection of java and json files contained in a folder that has a driver. The system will be operational after compiling all the components in the folder and running the program.

User Interface

Initial Display:

```
Welcome to the Code Cruncher learning management system!
```

1. Login
2. Create account

```
Please make your selection. :|
```

Account Creation:

```
Account Creation:
```

```
Please enter your first name : *****
```

```
Please enter your last name : *****
```

```
Please enter your date of birth in the form of Month/Day/Year : *****
```

```
Please enter your desired user name: *****
```

```
Please enter your chosen password: *****
```

```
Please enter your email address: *****@gmail.com|
```

Login:

```
*****
```

```
Login
```

```
Please enter your user name : *****
```

```
Please enter your password name : *****|
```

Student Main Menu:

```
*****
***** Student Main Menu *****
  1. View Courses
  2. View grades
  3. View certificates
  4. View awards
  5. Logout

Please make your selection. :|
```

Student View Courses Screen:

```
*****
***** Student - view courses *****

[Enter a number followed by the corresponding letter to open course]
[Enter 'E' to exit]

1. Python
   A- Course ***
   B- Course ****
2. JavaScript
   A- Course ***
   B- Course *****
   C- Course **
3. C#
   A- Course ****

Enter:
```

Student Lesson Menu :

```
*****
***** Student - view courses *****

[Enter 'E' to exit]

-----C#--COURSE **** -----

[1] Swapping variable values
[2] Boolean sort
[3] Arrays

Select a lesson: |
```

Student Take Lesson Screen:

```
*****
***** Student - C# Arrays *****

[Enter 'E' to exit]

-----C#--COURSE **** -----

1. You can initialise an array in c# using: int array[5] = {1,2,3,4,5}; where
   - 'array' is the identifier (name)
   - 'int' is the array type
   - '{}' denote the values inside the array

2. A 2 dimensional array is read rows by columns. For example,

   ****
   ****
   The above array is read as a 2x4 array!

[Enter 'N' to take the end of lesson quiz]

Enter:
```

Student Quiz:

```
*****
***** Student - COURSE ****
[Enter 'E' to exit]
-----C#--COURSE **** -----
1. The correct syntax for initialising an array in c# is:
[1] int a[3]={1,2,3};
[2] int a = {1,2,3};
[3] int a = new int[3]
[4] int a(3)= [1,2,3];
Answer: *
2. What array is this?
*****
*****
*****
[1] 4 x 3
[2] 3 x 6
[3] 6 x 3
[4] 3 x 3
Answer: *
3. In C# the length of an array cannot be changed. After it is instantiated with a length, dynamic resizing is not an option.
[1] True
[2] False
Answer: *
RESULT 2/3
3. The correct answer is [1]!
[Enter 'R' to retry
'E' to exit]
Enter: |
```

Student Exam:

```
*****
***** Student - C# EXAM *****
[Enter 'E' to exit]
-----C#--EXAM-----
1. The correct syntax for outputting "Hello World" in c# is:
[1] print("Hello world");
[2] cout << "Hello world";
[3] System.out.println("Hello world");
[4] Console.WriteLine("Hellow world");
Answer: *
2. What array is this?
*****
*****
*****
[1] 4 x 7
[2] 3 x 3
[3] 6 x 2
[4] 3 x 8
Answer: *
3. You can declare an array without dimensioning its size, but the size must be determined before you can reference it.
[1] True
[2] False
Answer: *
RESULT 3/5
You got it!
[Enter 'E' to claim certificate]
Enter: |
```

Student Grades Menu:

```
*****
***** Grades *****
Following are your current grades:
Python
- Course *** : A
- Course **  : C
JavaScript
[No courses completed]
C#
-Course **** : B
-Course **   : B
Go back? [Y/N] : |
```

Student Certificate Menu:

```
*****
***** Certificates *****
You have earned the following certificates:
    Completion of Python courses certificate
    Completion of JavaScript courses certificate
    Completion of C# courses certificate
Go Back? (Y/N) : |
```

Student Awards Menu:

```
*****
***** Awards *****
You have earned the following Awards:
    Badges: Code Master, Code Chef, Quick Learner
    Trophies: 100% Platinum Trophy, Crimson Trophy
Go Back? (Y/N) : |
```

Teacher Main Menu:

```
*****
***** Teacher Main Menu *****
1. View courses
2. Add lesson
3. Logout
```

```
Please make your selection. :|
```

Teacher View Courses:

```
*****
***** Teacher - view courses *****
1. Python
  - Course ***
  - Course ****
2. JavaScript
  - Course ***
  - Course *****
  - Course **
3. C#
  - Course *****
Go back [Y/N] :
```


Teacher Add Lesson:

```
*****
***** Teacher - Add lesson *****

Select from one of the following actions:
[1] Create fill-in-the-blank
[2] Create multiple choice question

Go back? [Y/N] :
```

Teacher Create Fill In the Blank:

```
*****
***** Teacher - Question creation *****

[Enter 'N' to roll back to previous menu]

Enter '_' in your question to create 'blanks'
Create Question: ****_**_*****

[Question created]

*****
[After filling each blank enter 'R' to redo previous blank
'E' to save and exit question creation]
Fill blank [1]: *****
Fill blank [2]: **
```

Activate Windows
Go to Settings to activate Windows.

Ln 15, Col 19

170%

Windows (CRLF)

UTF-8

Teacher Create Multiple Choice:

```
*****
***** Teacher - Question creation *****

[Enter 'N' to roll back to previous menu]

Create Question: *****

[Question created]
*****
[After each choice enter 'R' to redo previous choice
                          'Y' to create a new choice
                          'E' to save and exit question creation]

Create choice [1]: ***
Create choice [2]: *
```

Admin Main Menu:

```
*****
***** Admin Main Menu *****
1. View users
2. Remove user
3. Logout
```

Please make your selection. :|

Admin View Users:

```
*****
***** View users *****

Current users:
[1] Xxgoatedxx0
[2] freeicecream
[3] DehydratedHuman
[4] aihsdxfbqw
[5] 70coffee_man

Go back? [Y/N] : |
```

Admin Remove Users:

```
*****
***** Admin - Remove user *****

[Entries seperated by ',' will remove multiple users]
[Enter 'E' to exit]

[1] Xxgoatxx0
[2] freeicecream
[3] DehydratedHuman
[4] aihsdxfbqw
[5] 70coffee_man

Remove users: |
```

Log Out Screen:

```
*****  
***** LOGOUT *****  
Are you sure you would like to logout? [Y/N] :
```

Incorrect Input:

```
*****  
***** Incorrect Input Screen *****  
You have entered incorrectly formatted input.  
Please correct it to the following form: *****|
```

Data Storage

We will store the data for our system in 3 JSON files. One JSON file will be used to store and track all the users and their respective data including their name and usernames. The second JSON file will be used to store and track all the grades for each user. The third JSON file keeps track of the additional courses added to the system by the teachers.

Users.json:

```
1  [  
2    {  
3      "firstName": "LeBron",  
4      "lastName": "James",  
5      "DoB": "12/30/1984",  
6      "userName": "KingJames",  
7      "userPassword": "Ohio",  
8      "userEmail": "BronBron@gmail.com",  
9      "certificates": ["Python", "JavaScript","C#"]  
10   }, {  
11     "firstName": "Shaggy",  
12     "lastName": "Roggers",  
13     "DoB": "09/13/1969",  
14     "userName": "TheHunger",  
15     "userPassword": "Scooby Doo",  
16     "userEmail": "Shagman@gmail.com",  
17     "certificates": ["Python", "JavaScript","null"]  
18   }  
19 ]
```

Lessons:

```
1  [
2  {
3    "Course": [
4      {
5        "Course_type": "Python",
6        "Num_of_Lessons": "5",
7        "Teacher_First_Name": "Alice",
8        "Teacher_Last_Name": "Burrow",
9        "Teacher_DOB": "03/31/1980",
10       "Teahcer_email": "ABurrow@gmail.com",
11       "teacher_ID": "6ebc0a07-6105-4732-8cbb-e59d2e6f444d",
12       "Question": "How Do You Declare a Variable In Python?",
13       "Answer": "There is no command. It's created the moment you assign a value to it"
14     }
15   ]
16 },
17 {
18   "Course": [
19     {
20       "Course_Type": "JavaScript",
21       "Num_of_Lessons": "9",
22       "Teacher_First_Name": "James",
23       "Teacher_Last_Name": "Wright",
24       "Teacher_DOB": "10/20/1992",
25       "Teahcer_email": "WrightJ@gmail.com",
26       "teacher_ID": "2023a887-ed0f-4a46-bca9-4bb2e87b162e",
27       "Question": "How Do You Declare a Variable In JavaScript?",
28       "Answer": "Var x = whatever, etc"
29     }
30   ]
}
```

Grades:

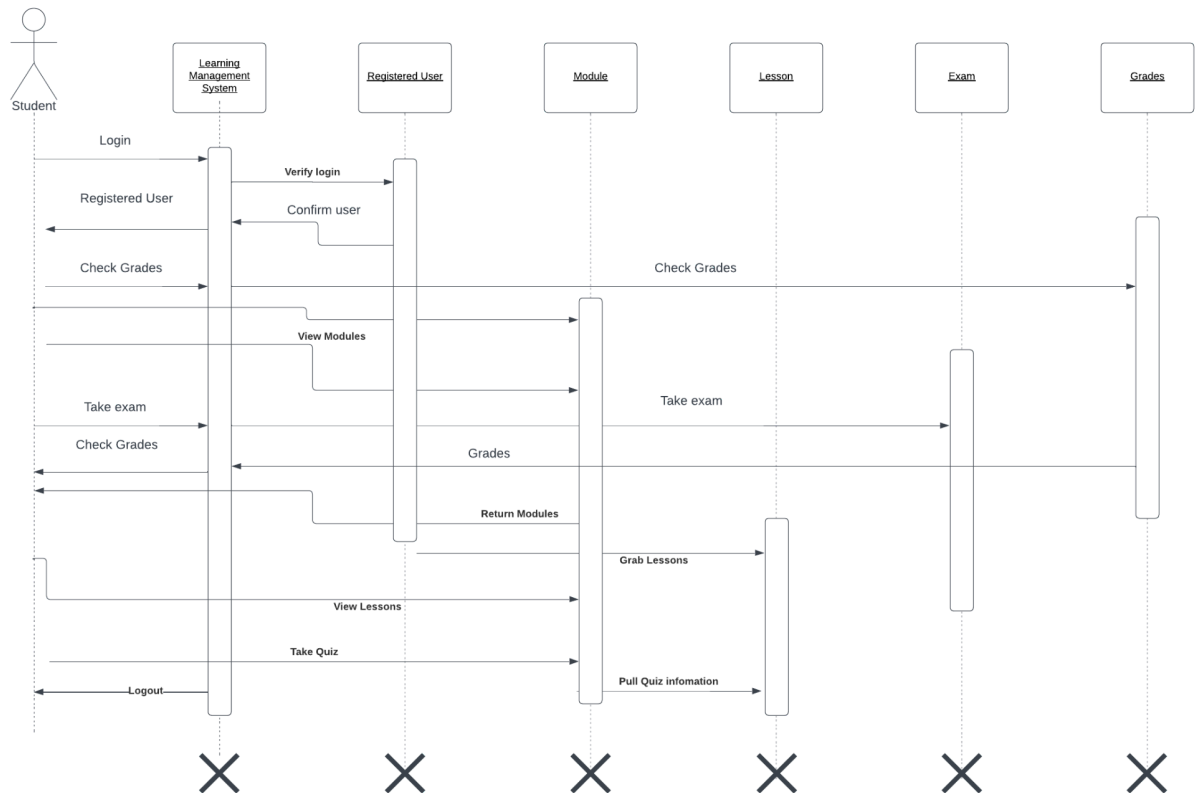
```
1  [
2  {
3      "LeBron James": [
4          {
5              "Lesson": "Python",
6              "grades": {
7                  "course": [
8                      {
9                          "course_title": "Lesson 1",
10                         "int_grade": "",
11                         "ext_grade": "",
12                         "letter_grade": "B",
13                         "Date_of_Exam": "05/2023",
14                     }
15                 ]
16             }
17         }
18     }
19 ]
```

Class Diagram – Structural Design

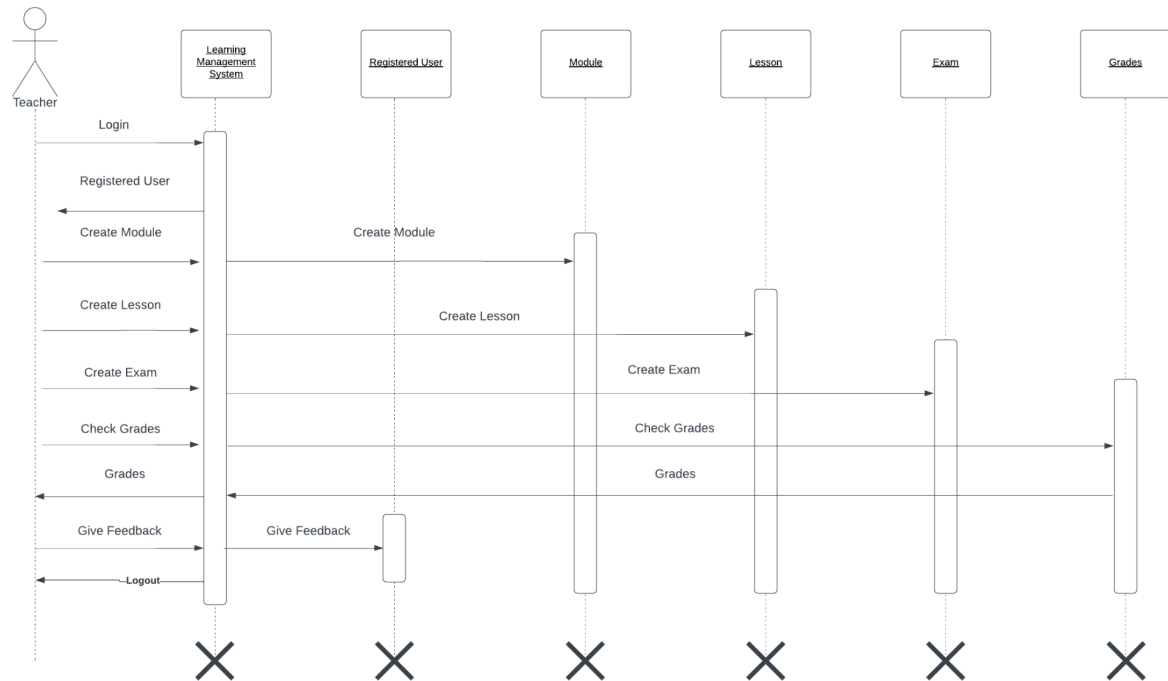
Please see the PDF

Sequence Diagrams - Dynamic Model

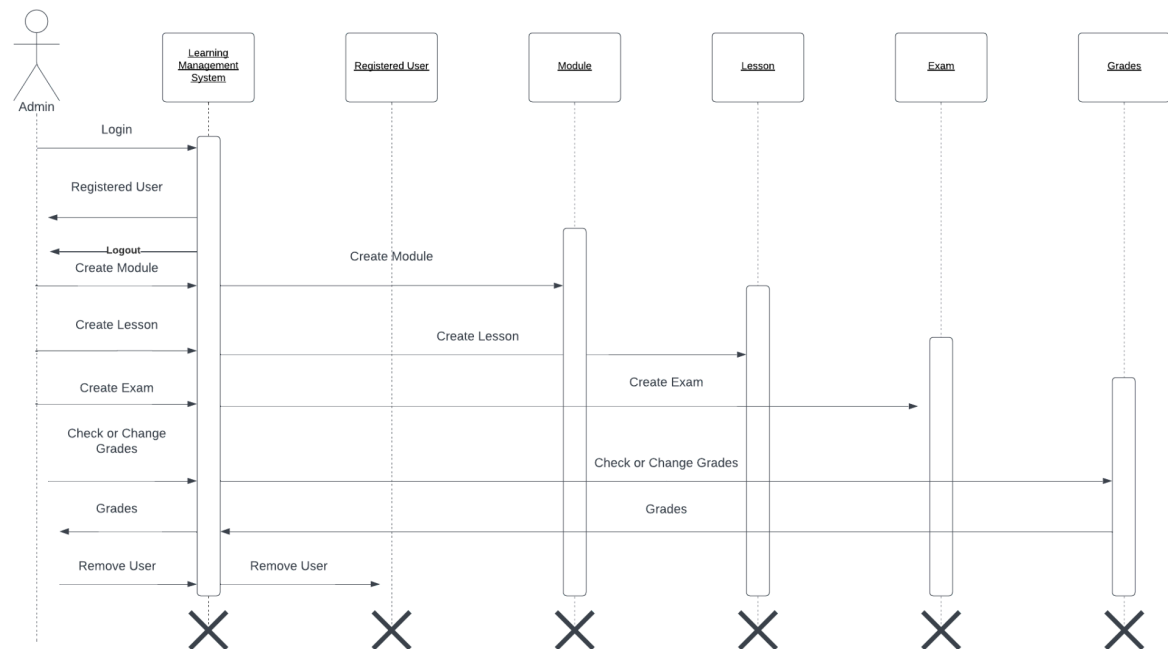
Student Sequence Diagram:



Teacher Sequence Diagram:



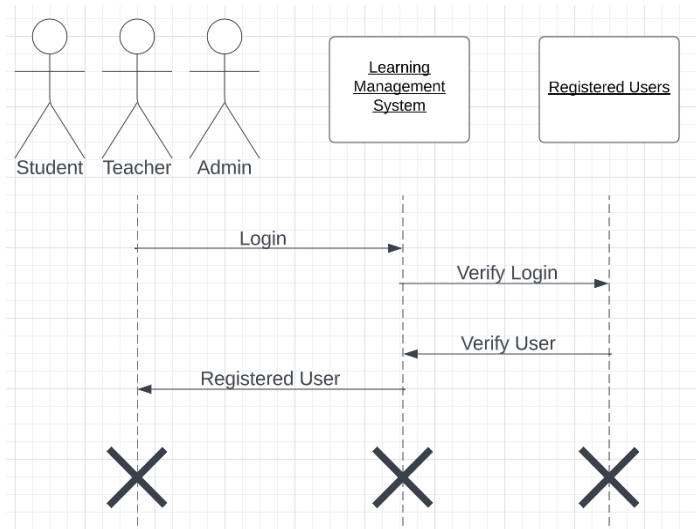
Admin Sequence Diagram



Scenarios

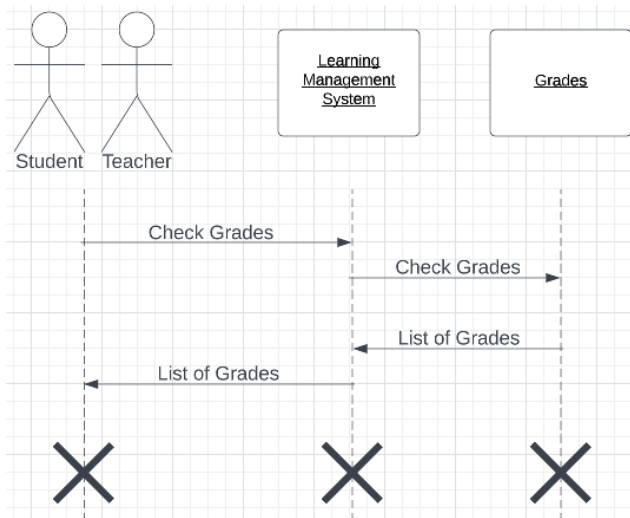
Scenario #1: User Login and Logout

- **Scenario Name:** Login and Logout
- **Scenario Description:** The general user first logs into the system, and then the system checks if the user is registered or not. If the user is registered, the system will allow the user in. Then, the user can logout from the main menu whenever they please.
- **Sequence Diagram:**



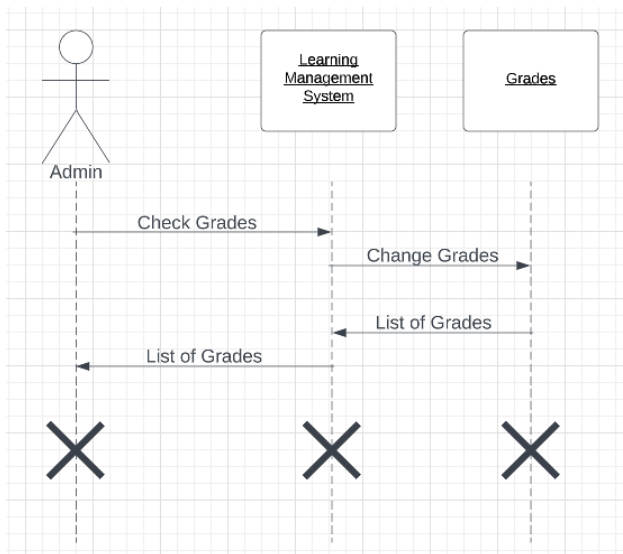
Scenario #2: Student and Teacher viewing grade

- **Scenario Name:** Viewing Grade(s)
- **Scenario Description:** The student can look at their grades from the main menu. The teacher can look at the grades for the students as well.
- **Sequence Diagram:**



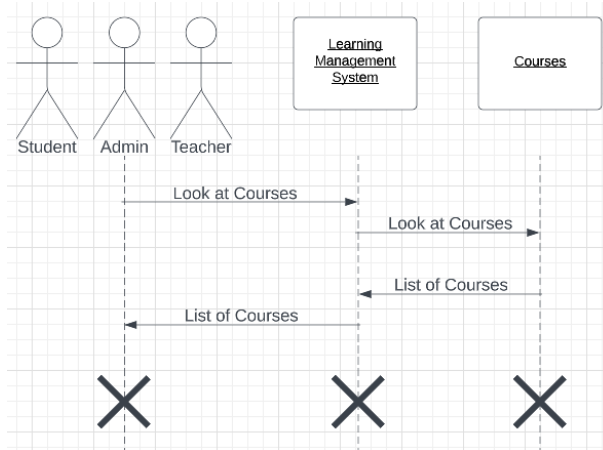
Scenario #3: Admin viewing or changing Grade(s)

- **Scenario Name:** Changing Grade(s)
- **Scenario Description:** The admin can look, remove, or change student grades from the main menu.
- **Sequence Diagram:**



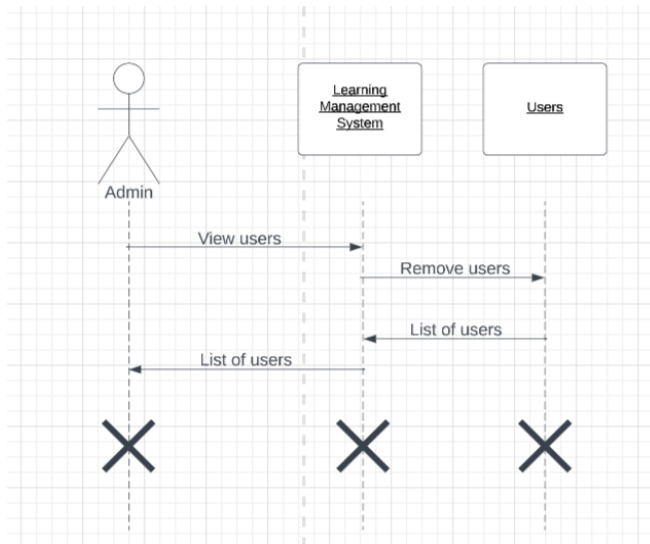
Scenario #4: User viewing Course(s)

- **Scenario Name:** Viewing Course(s)
- **Scenario Description:** The general user can view the list of courses from the main menu.
- **Sequence Diagram:**



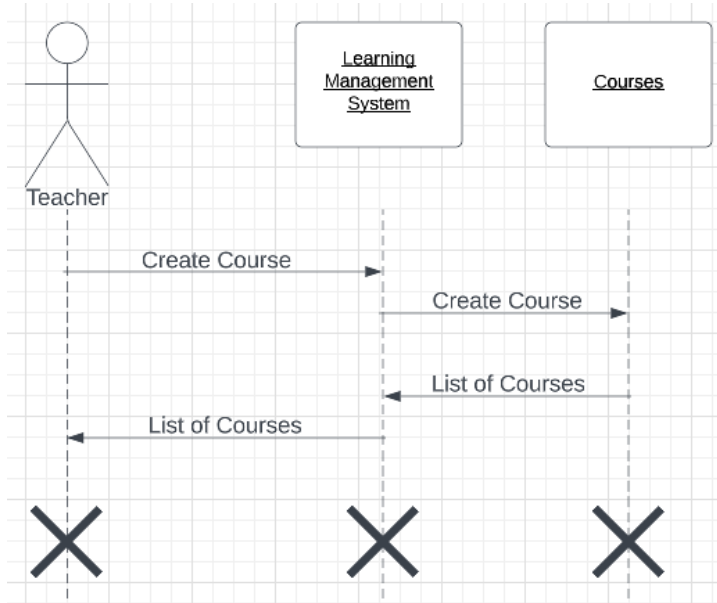
Scenario #5: Admin removing User(s)

- **Scenario Name:** Removing User(s)
- **Scenario Description:** The admin can look at a list of registered users, and remove a user of their choice.
- **Sequence Diagram:**



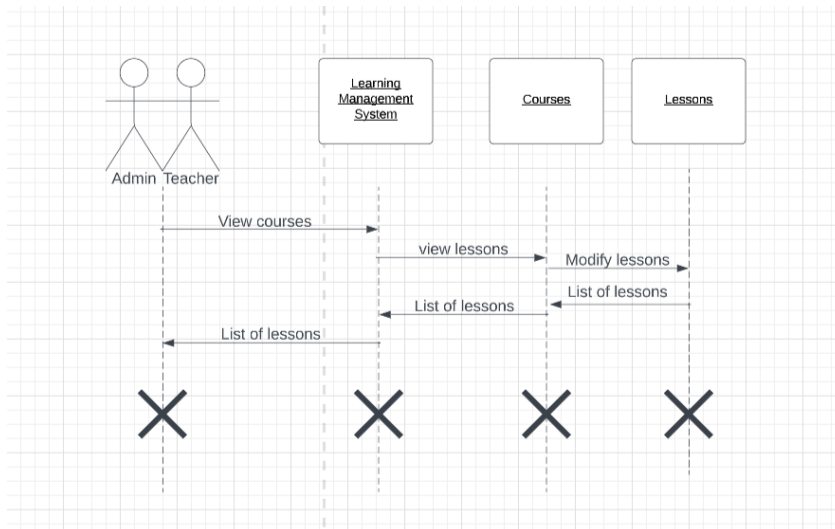
Scenario #6: Teacher creating course(s)

- **Scenario Name:** Creating course(s)
- **Scenario Description:** The teacher can select from multiple languages to create a course for that language type
- **Sequence Diagram:**



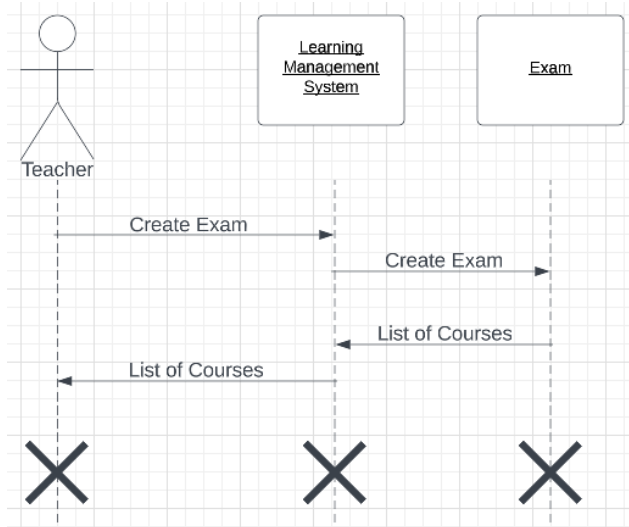
Scenario #7: Teacher or Admin adding and removing Lesson(s)

- **Scenario Name:** Removing Lesson(s)
- **Scenario Description:** The admin can choose from multiple lessons to remove
- **Sequence Diagram:**



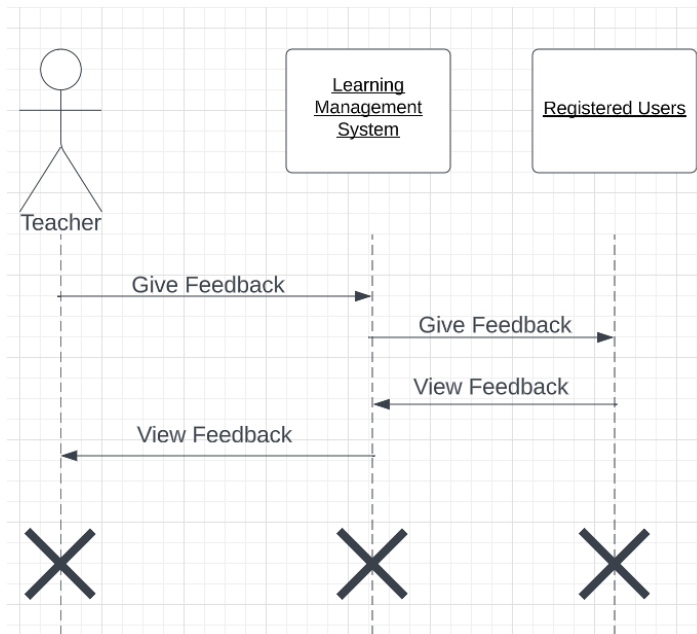
Scenario #8: Teacher adding Exam(s)

- **Scenario Name:** Adding Exam(s)
- **Scenario Description:** The teacher can add a final exam for a particular course
- **Sequence Diagram:**



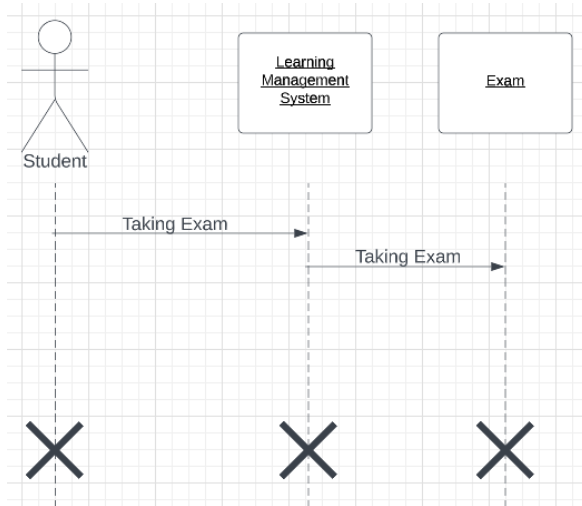
Scenario #9: Teacher giving Feedback

- **Scenario Name:** Giving Feedback
- **Scenario Description:** The teacher can give feedback to students on how they could improve based on their grades.
- **Sequence Diagram:**



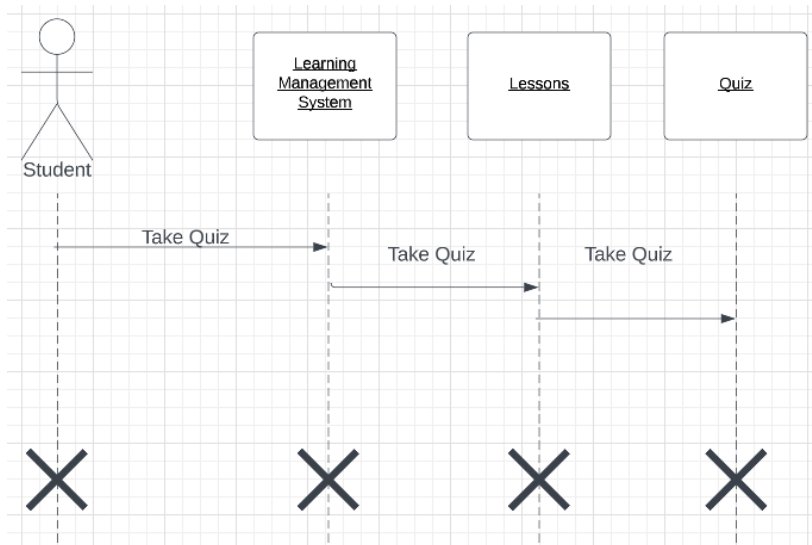
Scenario #10: Student taking Exam(s)

- **Scenario Name:** Taking Exam(s)
- **Scenario Description:** The student can take an exam after having completed a course and all of the lessons it contains, in order to receive a certificate
- **Sequence Diagram:**



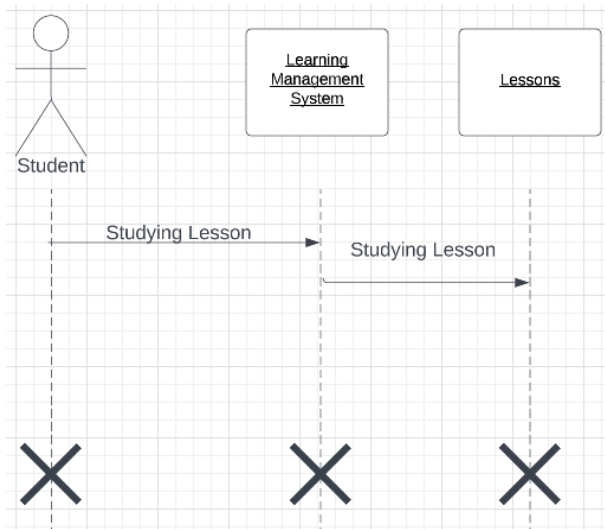
Scenario #11: Student taking Quizzes

- **Scenario Name:** Taking Quizzes
- **Scenario Description:** The student can take a quiz after having completed a lesson before moving on to the next lesson.
- **Sequence Diagram:**



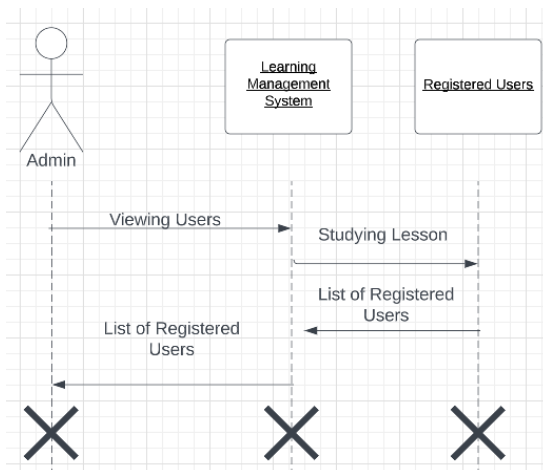
Scenario #12: Student taking Lesson(s)

- **Scenario Name:** Taking Lesson(s)
- **Scenario Description:** The student can take any lessons within a course
- **Sequence Diagram:**



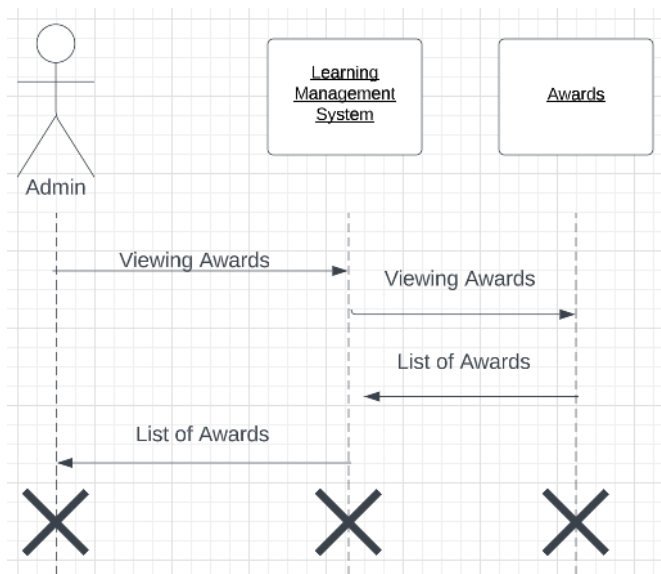
Scenario #13: Admin viewing users

- **Scenario Name:** View users
- **Scenario Description:** The admin may view a list of all registered users
- **Sequence Diagram:**



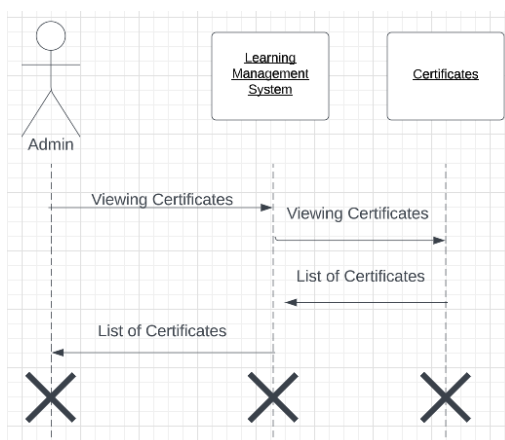
Scenario #14: Student viewing awards

- **Scenario Name:** view awards
- **Scenario Description:** The student can view all awards they have accumulated whilst using the program.
- **Sequence Diagram:**



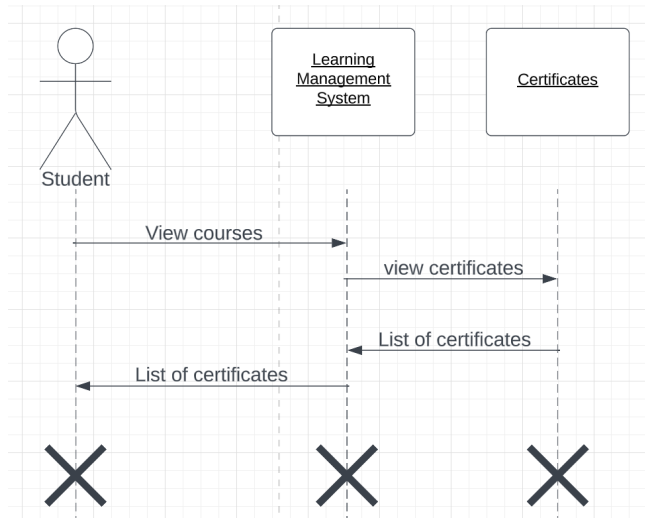
Scenario #15: Student viewing certificates

- **Scenario Name:** view certificates
- **Scenario Description:** The student can view all certificates they have accumulated whilst using the program.
- **Sequence Diagram:**



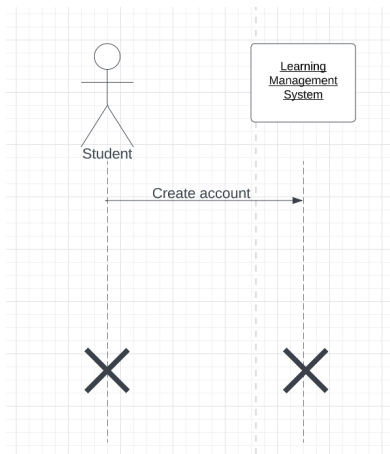
Scenario #16: Student receiving certificates

- **Scenario Name:** Receive certificate
- **Scenario Description:** The student can receive and accept a certificate after completion of a course for a particular language
- **Sequence Diagram:**



Scenario #17: A user creates an account

- **Scenario Name:** Account creation
- **Scenario Description:** The general user can create an account with all the required credentials and maintain this account
- **Sequence Diagram:**



Scenario #18: Student viewing feedback

- **Scenario Name:** Viewing Feedback
- **Scenario Description:** The student can view the feedback that the teacher gave to them from the main menu
- **Sequence Diagram:**

