Design Document

System Overview

The learning management system (LMS) is designed to provide learning material as well as assessments for students to develop coding skills in select coding languages. The system will allow authors to create course content that will be provided to those students. In addition, these courses may offer certificates to the students to show proof of knowledge gained from the course.

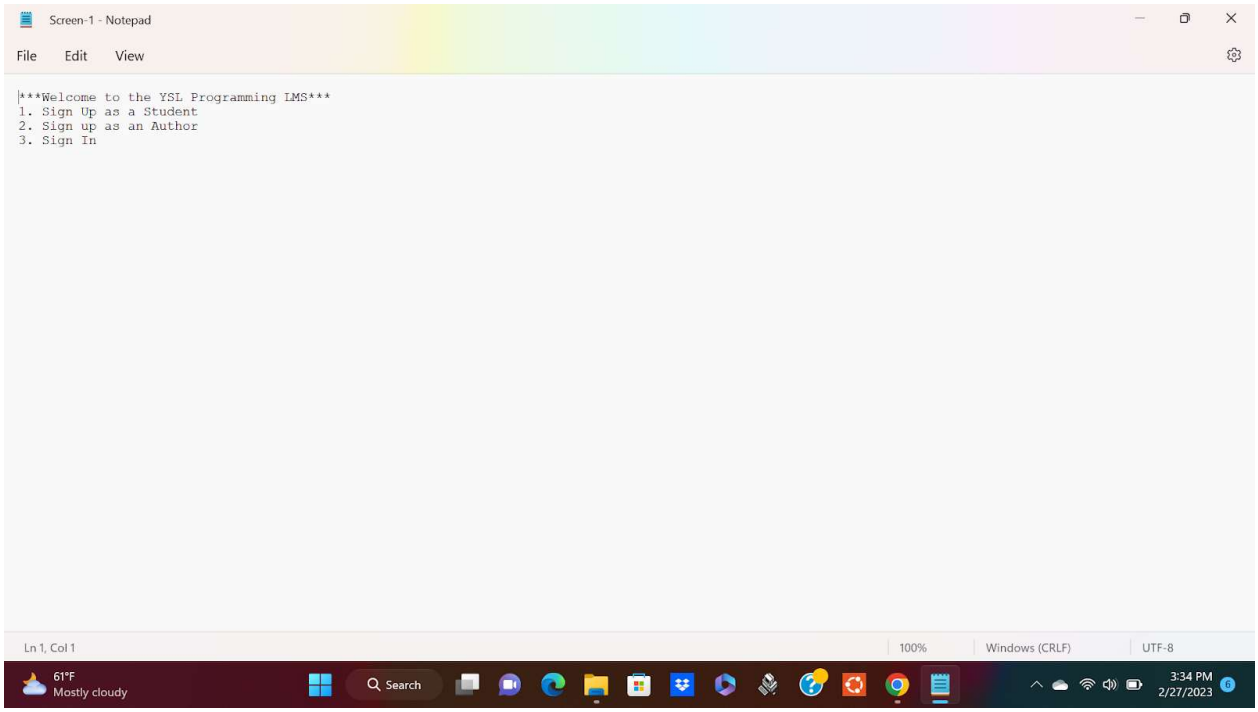
References

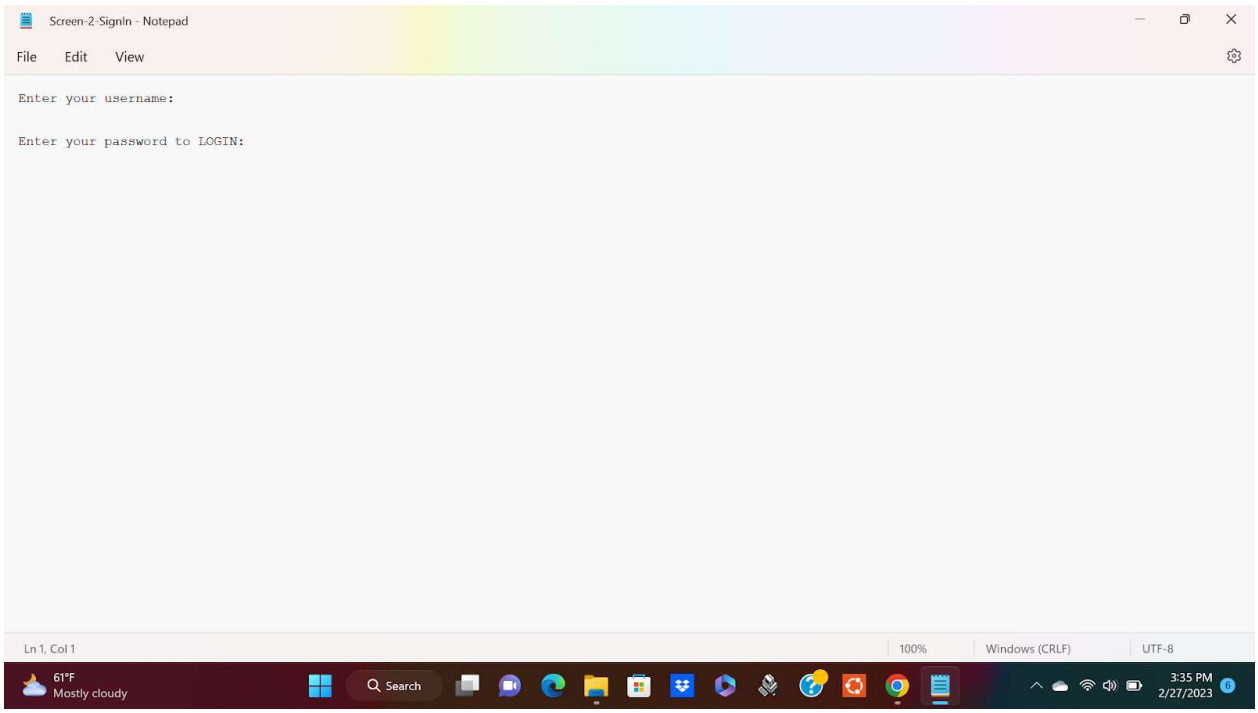
- LMS UML.pdf
- Sequence Diagrams.pdf
- Development Requirements.pdf
- users.json
- courses.json
- comment.json
- review.json

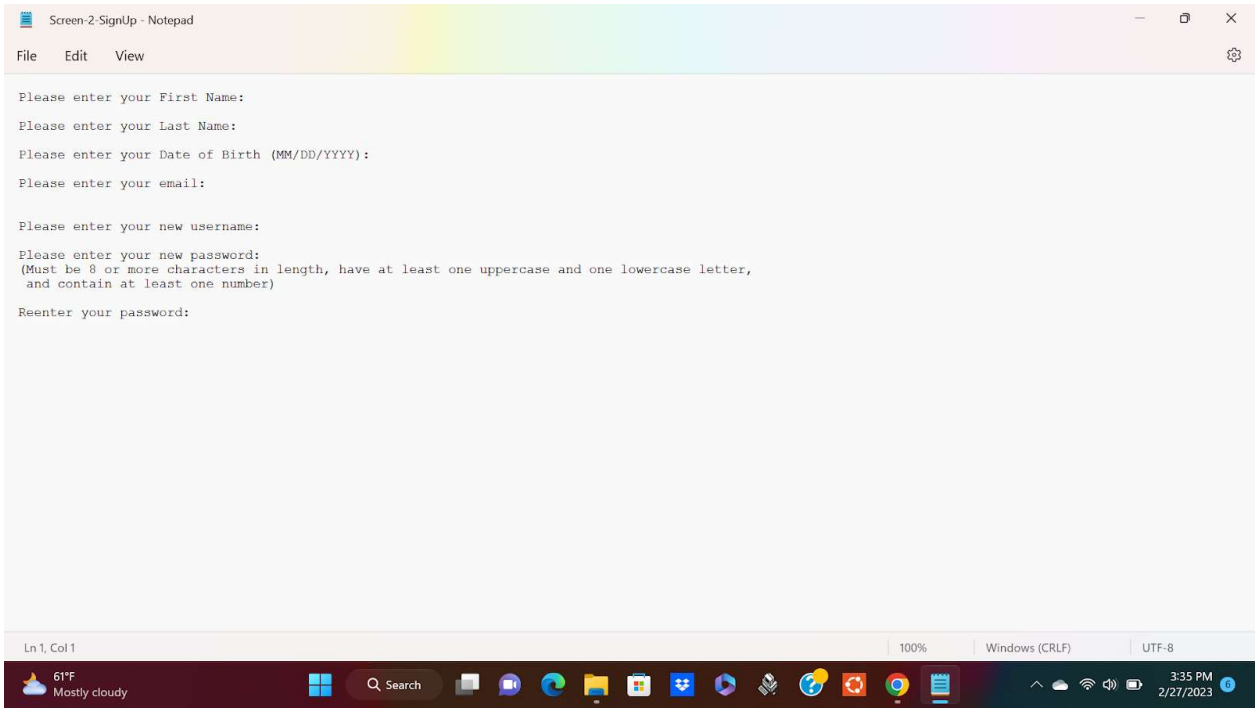
Environment Overview

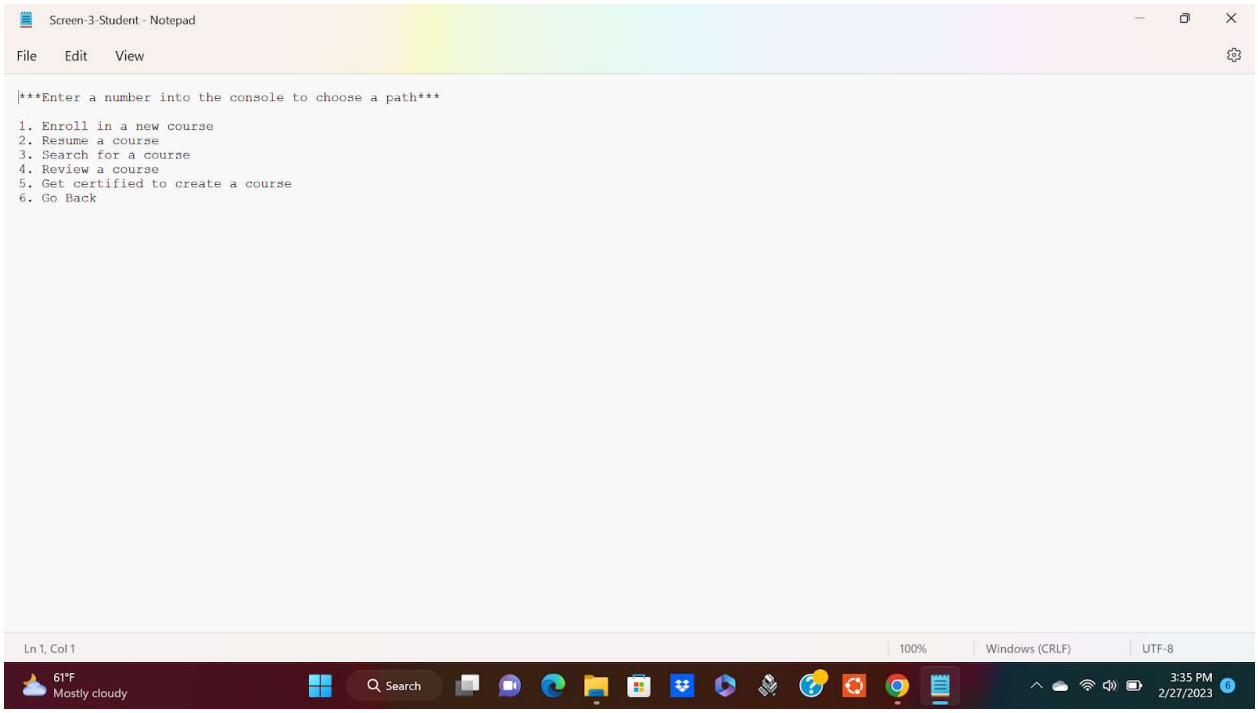
The LMS will run as a console application on the end user's computer. The program will also come with a users.json file as well as a courses.json file that must be in the same directory as the program executable file. These two files provide the registered users as well as the courses that are available. The program will be run by opening the executable file.

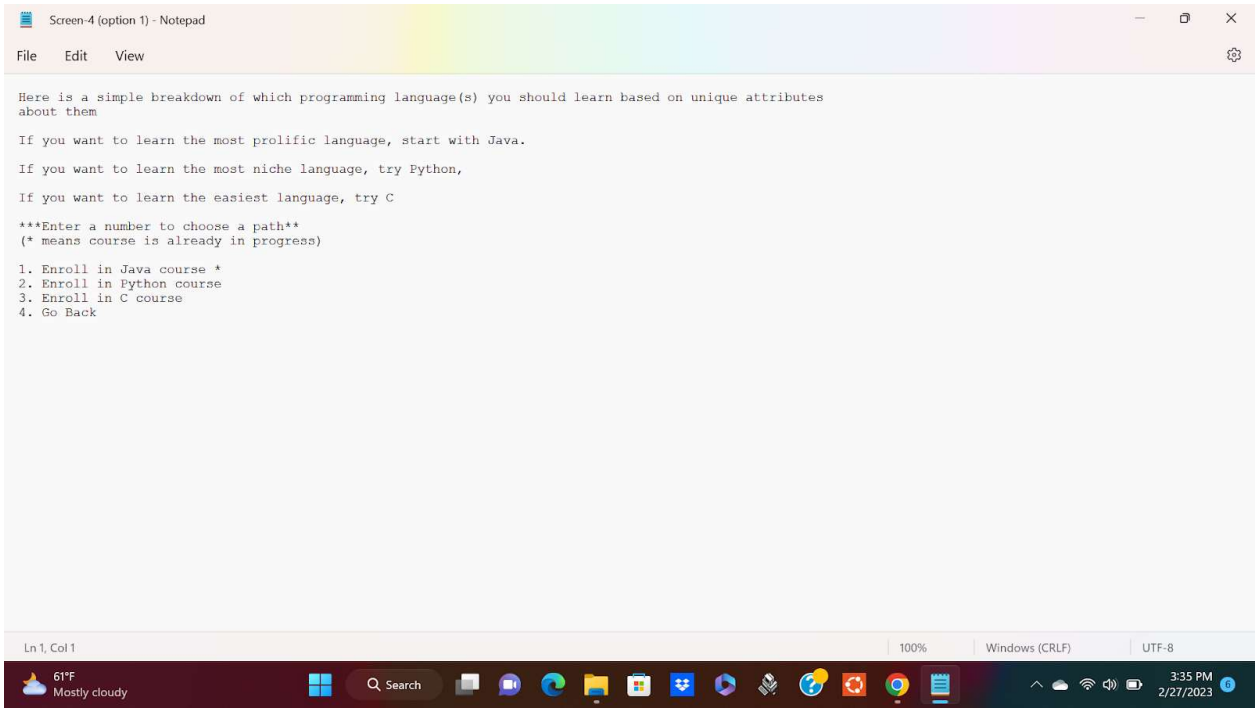
User Interface

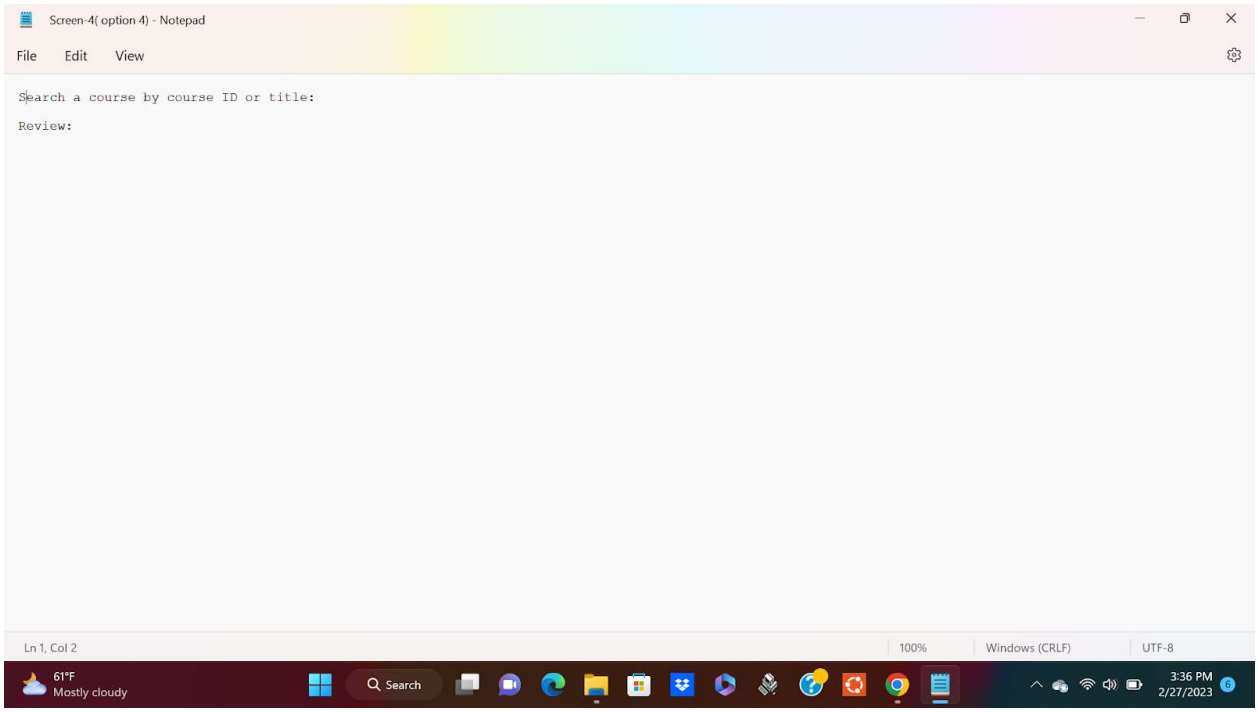

Screen-1 - Notepad
File Edit View
Welcome to the YSL Programming LMS
1. Sign Up as a Student
2. Sign up as an Author
3. Sign In
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:34 PM
2/27/2023

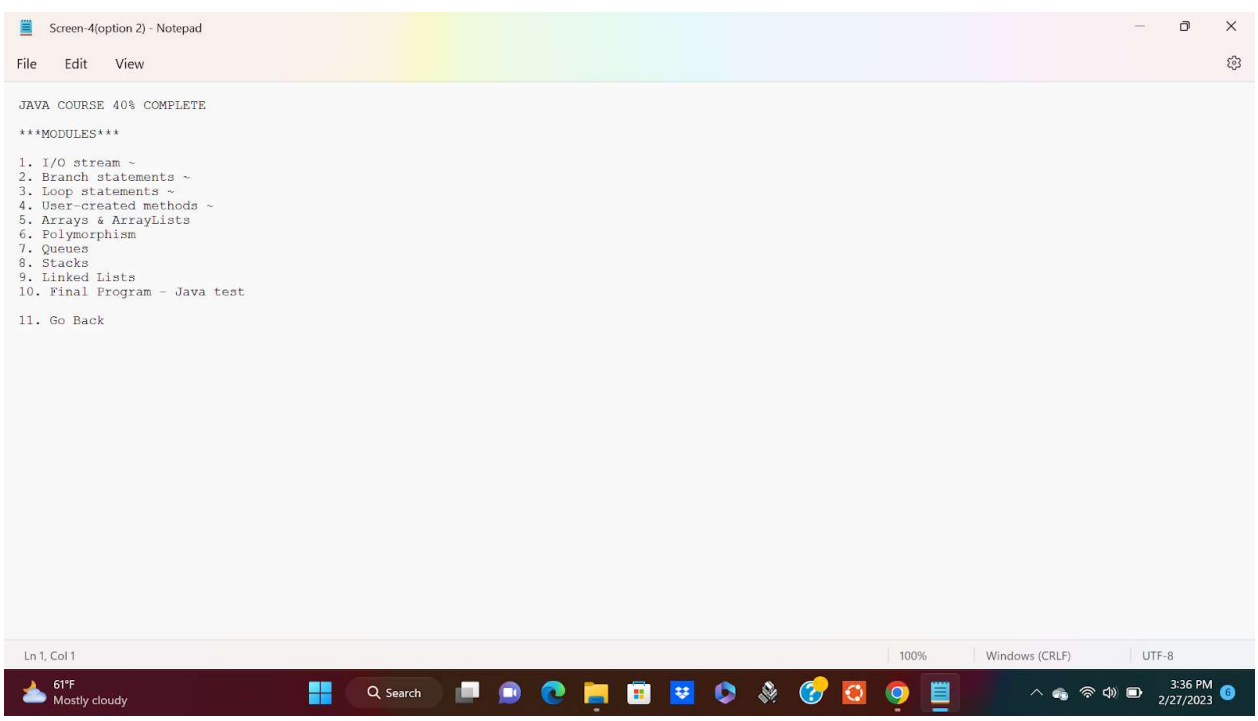

Screen-2-SignIn - Notepad
File Edit View
Enter your username:
Enter your password to LOGIN:
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:35 PM
2/27/2023

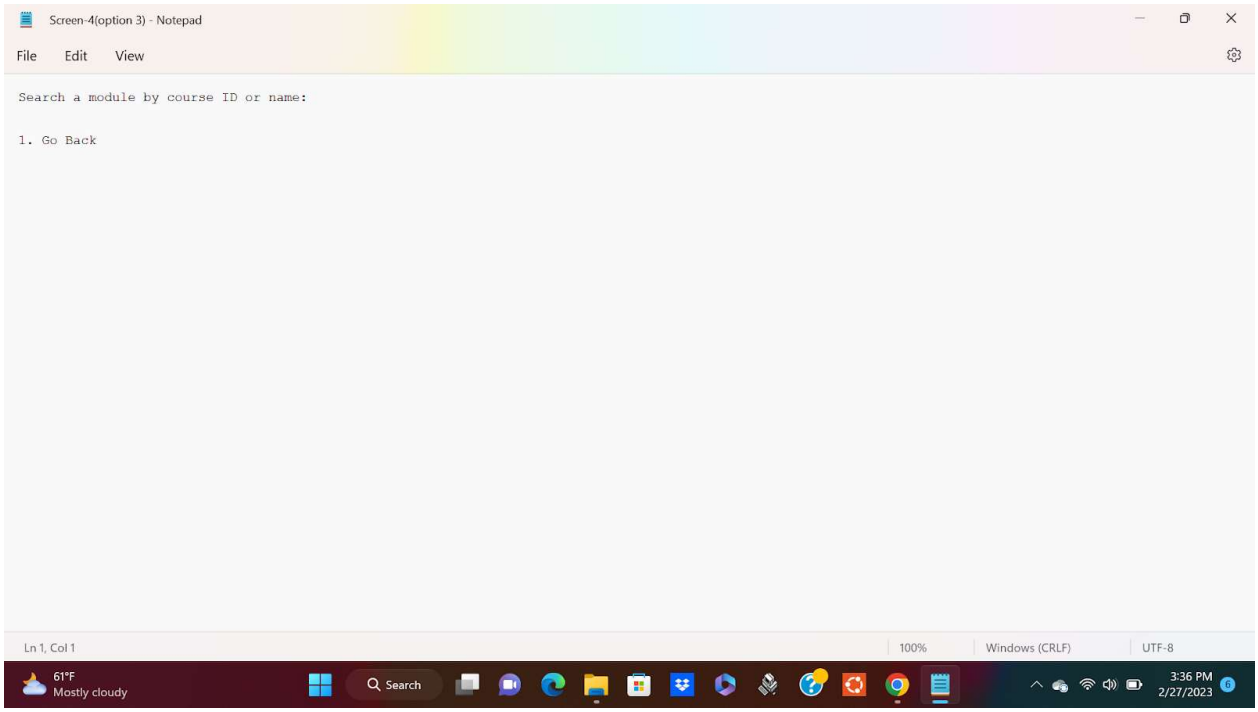

Screen-2-SignUp - Notepad
File Edit View
Please enter your First Name:
Please enter your Last Name:
Please enter your Date of Birth (MM/DD/YYYY):
Please enter your email:
Please enter your new username:
Please enter your new password:
(Must be 8 or more characters in length, have at least one uppercase and one lowercase letter,
and contain at least one number)
Reenter your password:
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:35 PM
2/27/2023

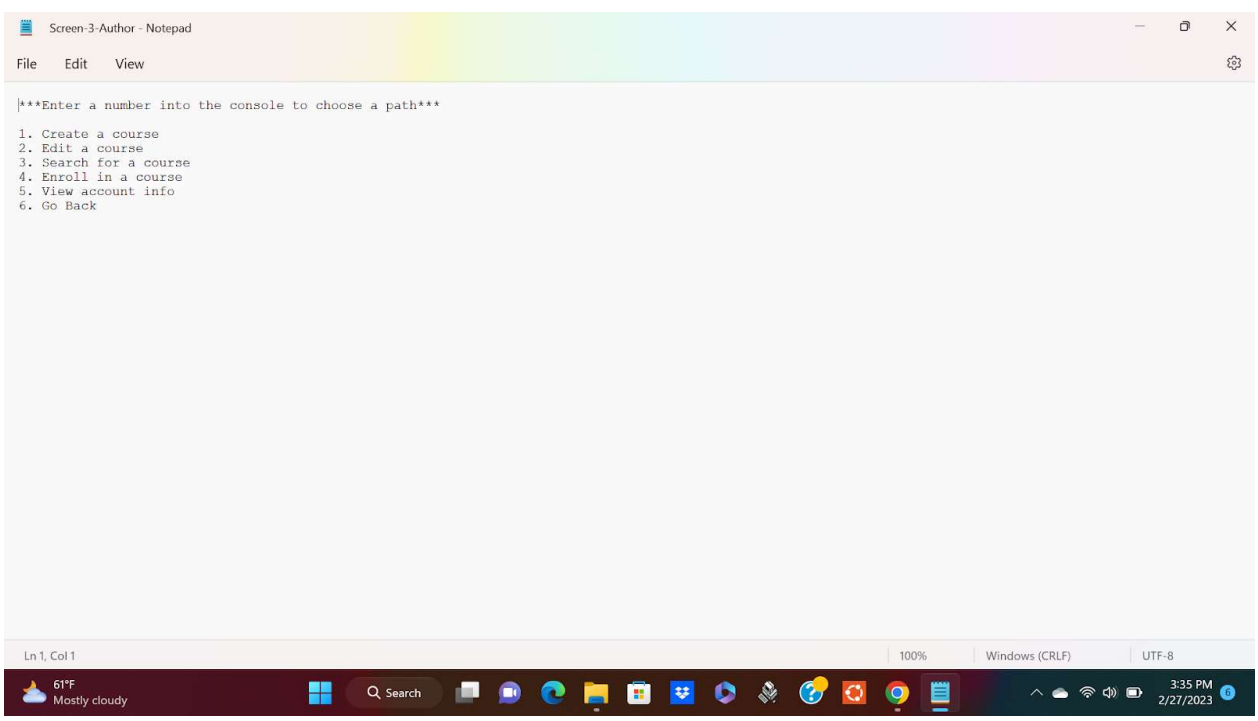

Screen-3-Student - Notepad
File Edit View
Enter a number into the console to choose a path
1. Enroll in a new course
2. Resume a course
3. Search for a course
4. Review a course
5. Get certified to create a course
6. Go Back
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:35 PM
2/27/2023

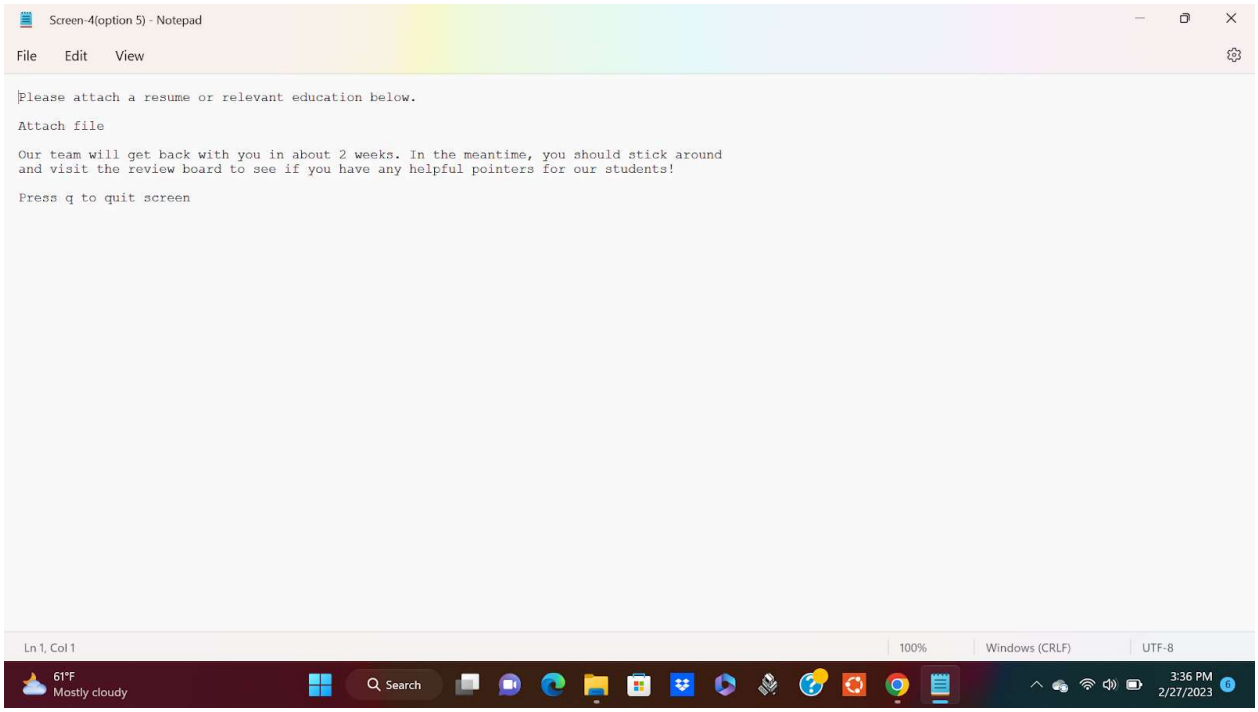

Screen-4 (option 1) - Notepad
File Edit View
Here is a simple breakdown of which programming language(s) you should learn based on unique attributes about them
If you want to learn the most prolific language, start with Java.
If you want to learn the most niche language, try Python,
If you want to learn the easiest language, try C
***Enter a number to choose a path**
(* means course is already in progress)
1. Enroll in Java course *
2. Enroll in Python course
3. Enroll in C course
4. Go Back
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:35 PM
2/27/2023

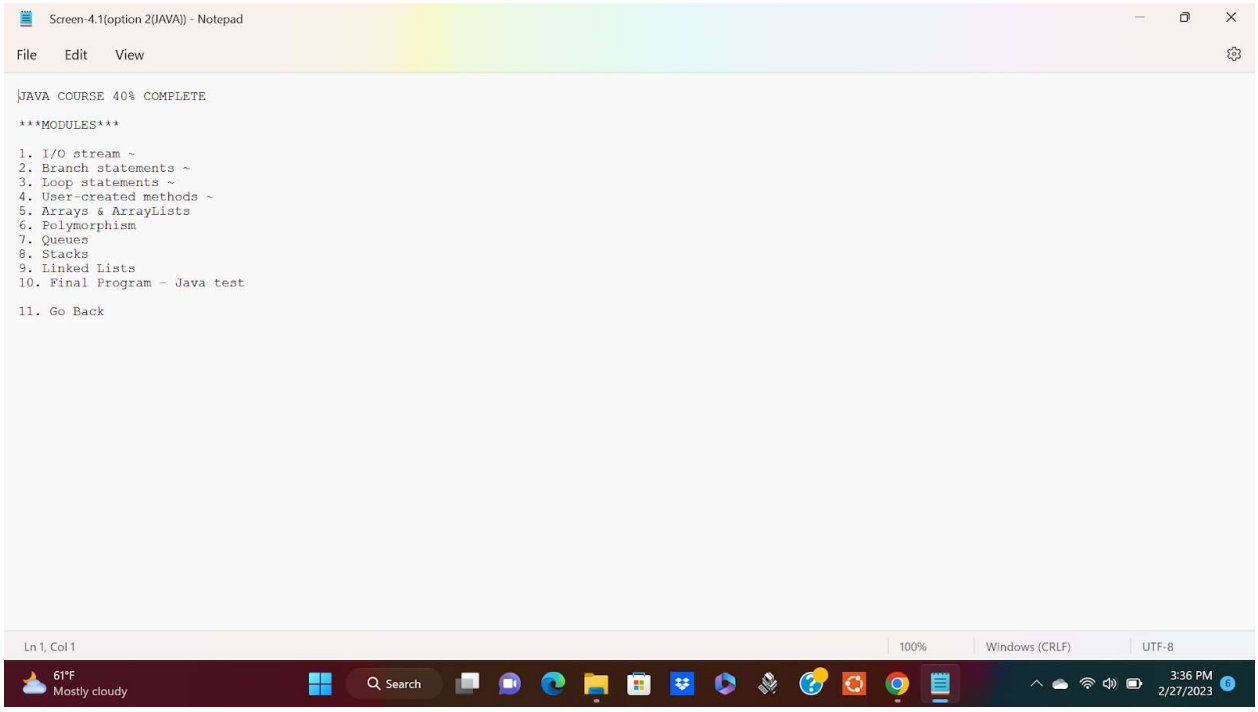

Screen-4(option 4) - Notepad
File Edit View
Search a course by course ID or title:
Review:
Ln 1, Col 2
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:36 PM
2/27/2023

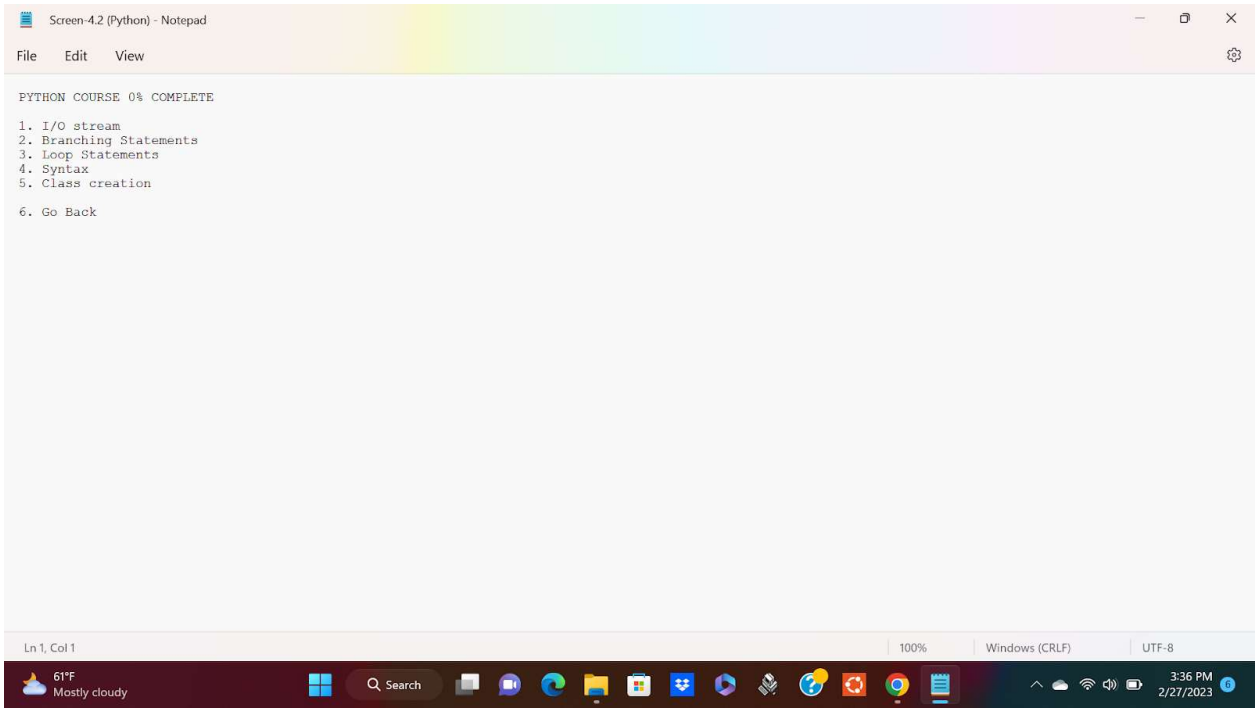

Screen-4(option 2) - Notepad
File Edit View
JAVA COURSE 40% COMPLETE
MODULES
1. I/O stream ~
2. Branch statements ~
3. Loop statements ~
4. User-created methods ~
5. Arrays & ArrayLists
6. Polymorphism
7. Queues
8. Stacks
9. Linked Lists
10. Final Program - Java test
11. Go Back
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:36 PM
2/27/2023

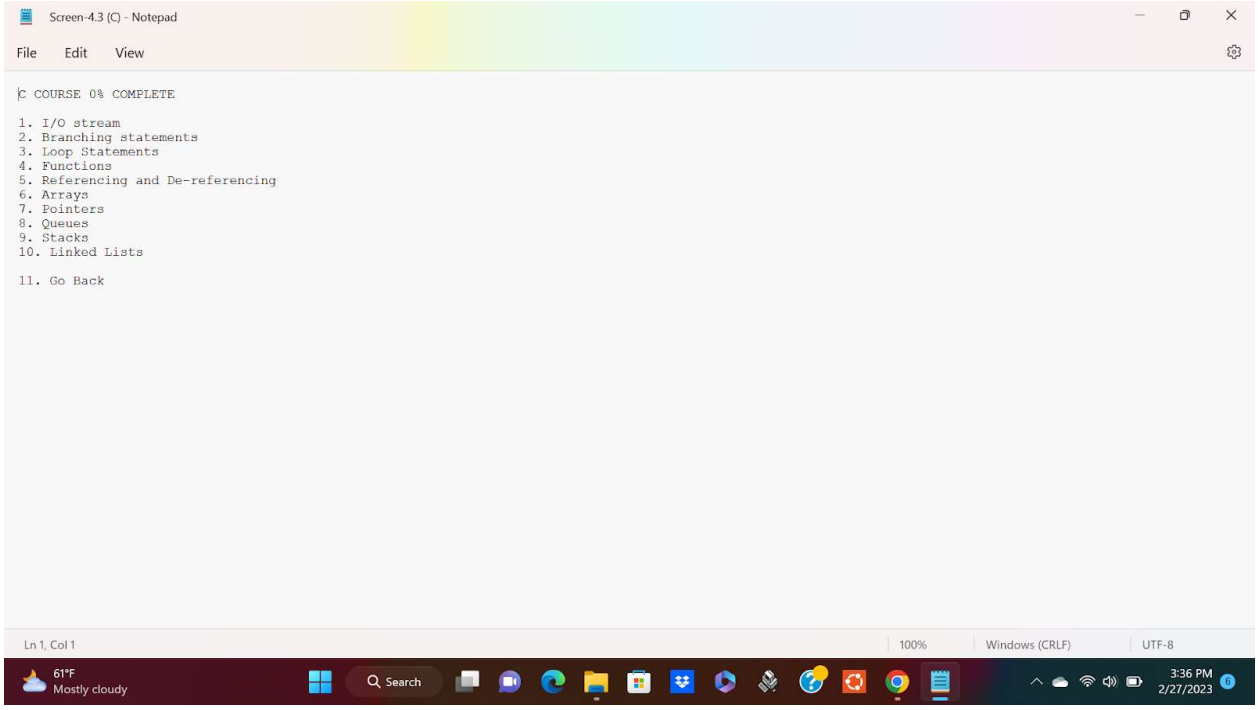

Screen-4(option 3) - Notepad
File Edit View
Search a module by course ID or name:
1. Go Back
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:36 PM
2/27/2023

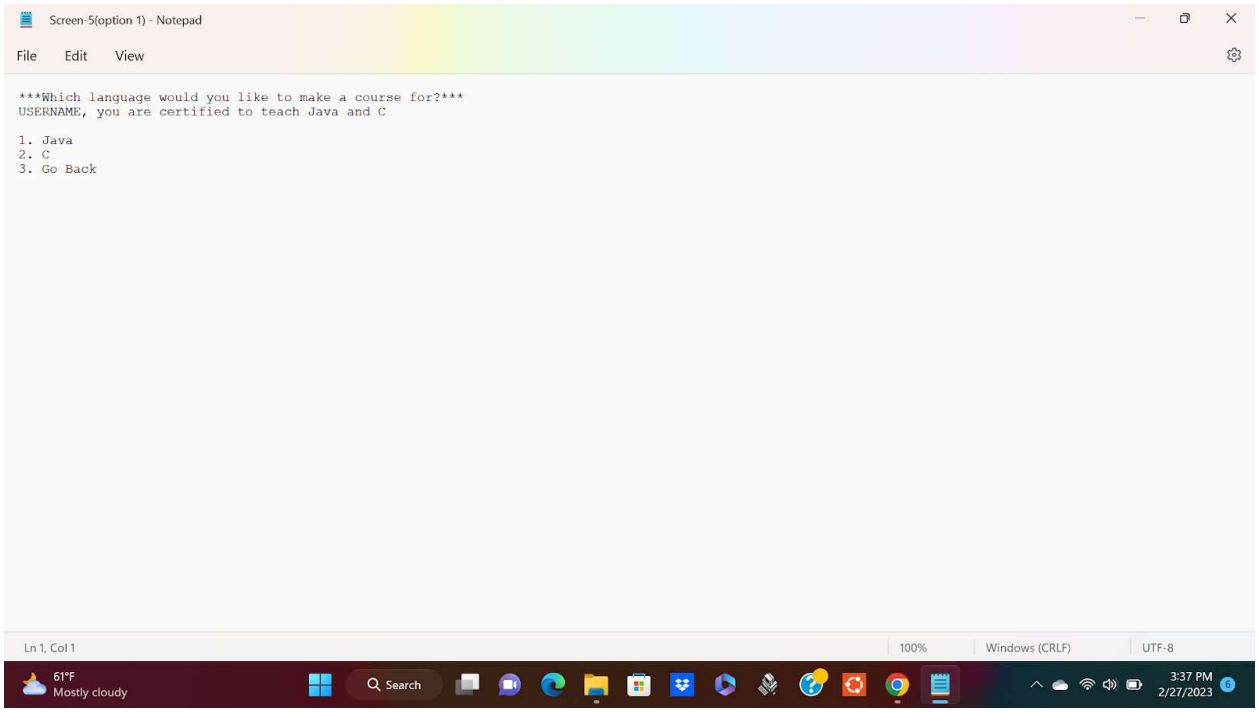

Screen-3-Author - Notepad
File Edit View
Enter a number into the console to choose a path
1. Create a course
2. Edit a course
3. Search for a course
4. Enroll in a course
5. View account info
6. Go Back
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:35 PM
2/27/2023

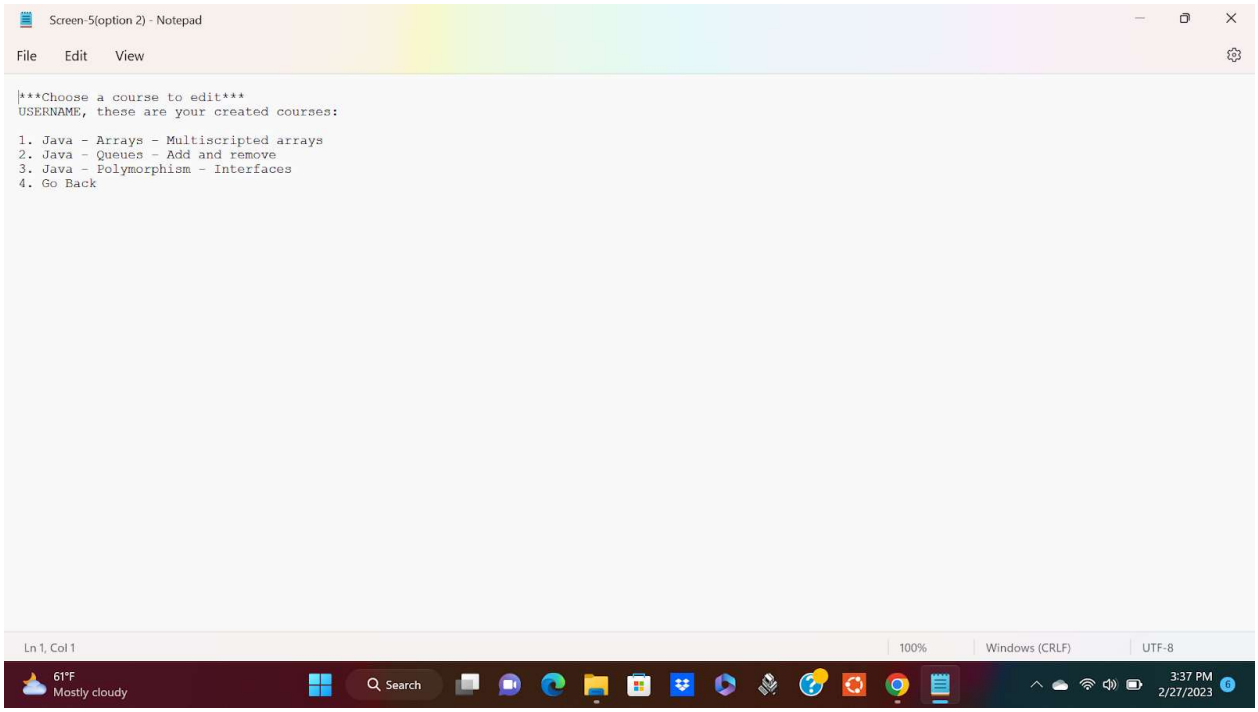

Screen-4(option 5) - Notepad
File Edit View
Please attach a resume or relevant education below.
Attach file
Our team will get back with you in about 2 weeks. In the meantime, you should stick around
and visit the review board to see if you have any helpful pointers for our students!
Press q to quit screen
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:36 PM
2/27/2023

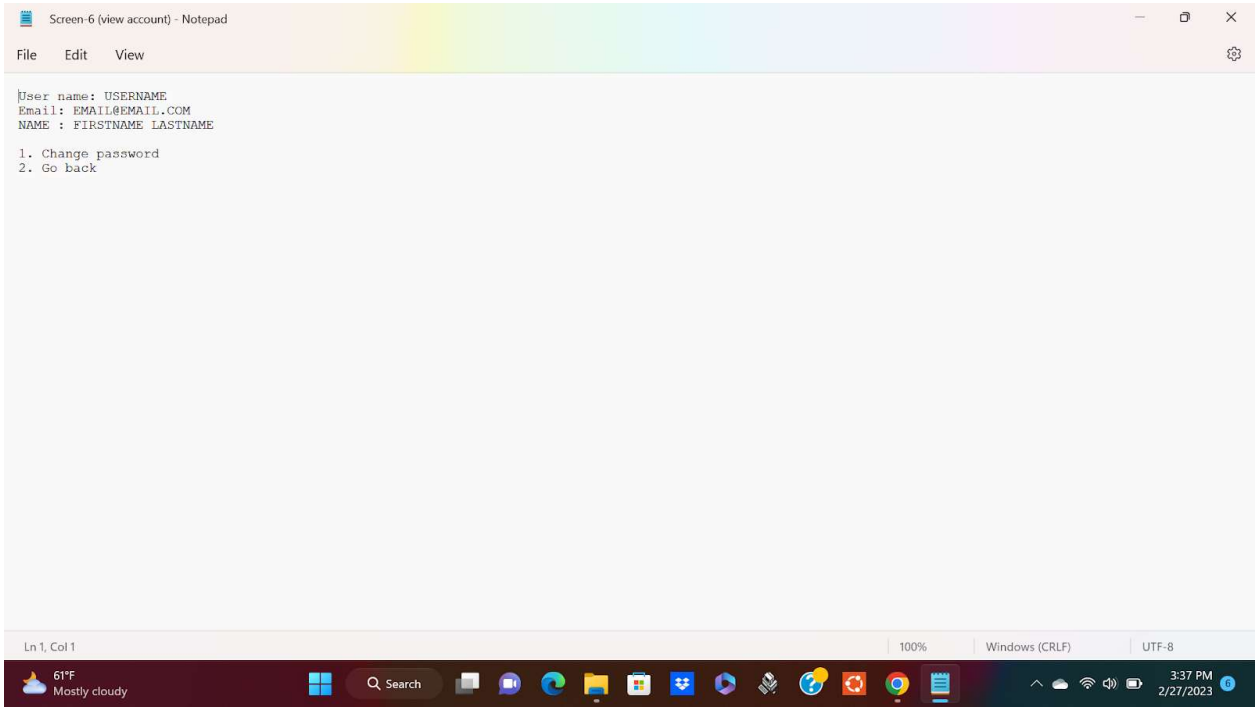

Screen-4.1(option 2(JAVA)) - Notepad
File Edit View
JAVA COURSE 40% COMPLETE
MODULES
1. I/O stream ~
2. Branch statements ~
3. Loop statements ~
4. User-created methods ~
5. Arrays & ArrayLists
6. Polymorphism
7. Queues
8. Stacks
9. Linked Lists
10. Final Program - Java test
11. Go Back
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:36 PM
2/27/2023

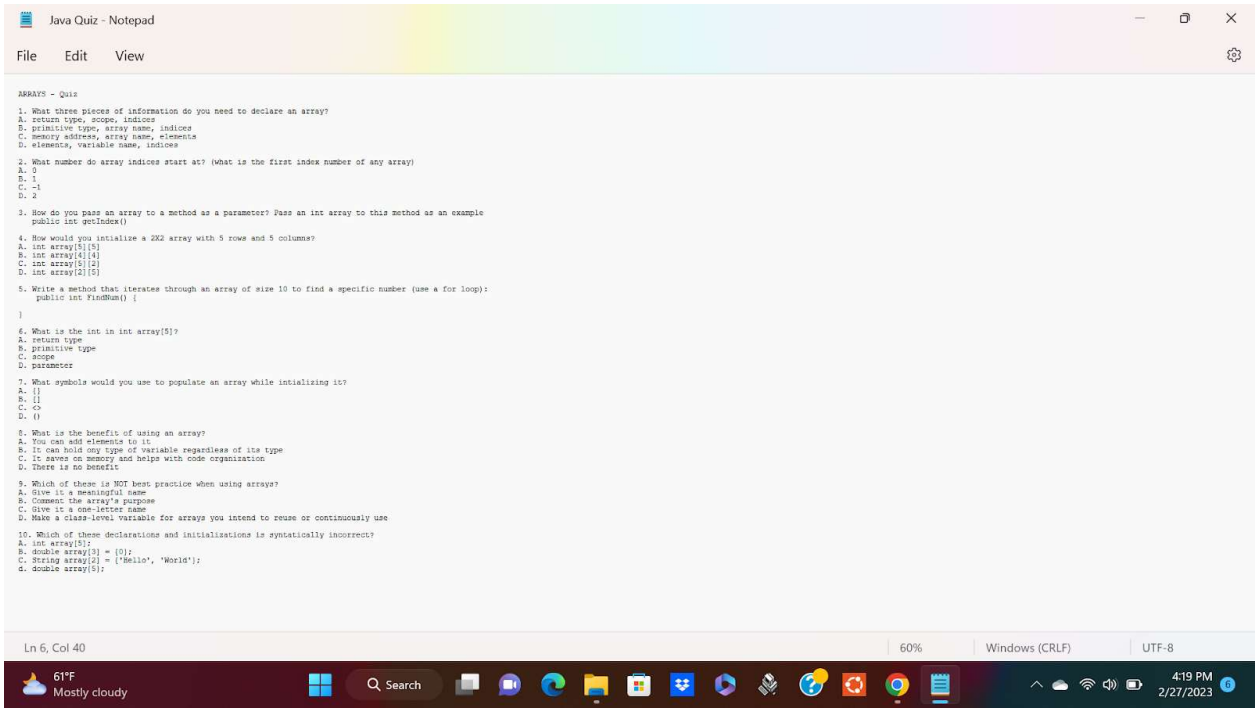

Screen-4.2 (Python) - Notepad
File Edit View
PYTHON COURSE 0% COMPLETE
1. I/O stream
2. Branching Statements
3. Loop Statements
4. Syntax
5. Class creation
6. Go Back
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:36 PM
2/27/2023


Screen-4.3 (C) - Notepad
File Edit View
C COURSE 0% COMPLETE
1. I/O stream
2. Branching statements
3. Loop Statements
4. Functions
5. Referencing and De-referencing
6. Arrays
7. Pointers
8. Queues
9. Stacks
10. Linked Lists
11. Go Back
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:36 PM
2/27/2023


Screen-5(option 1) - Notepad
File
Edit
View
Which language would you like to make a course for?
USERNAME, you are certified to teach Java and C
1. Java
2. C
3. Go Back
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:37 PM
2/27/2023


Screen-5(option 2) - Notepad
File Edit View
Choose a course to edit
USERNAME, these are your created courses:
1. Java - Arrays - Multiscripted arrays
2. Java - Queues - Add and remove
3. Java - Polymorphism - Interfaces
4. Go Back
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:37 PM
2/27/2023


Screen-6 (view account) - Notepad
File Edit View
User name: USERNAME
Email: EMAIL@EMAIL.COM
NAME : FIRSTNAME LASTNAME
1. Change password
2. Go back
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
61°F
Mostly cloudy
Search
3:37 PM
2/27/2023



Data Storage

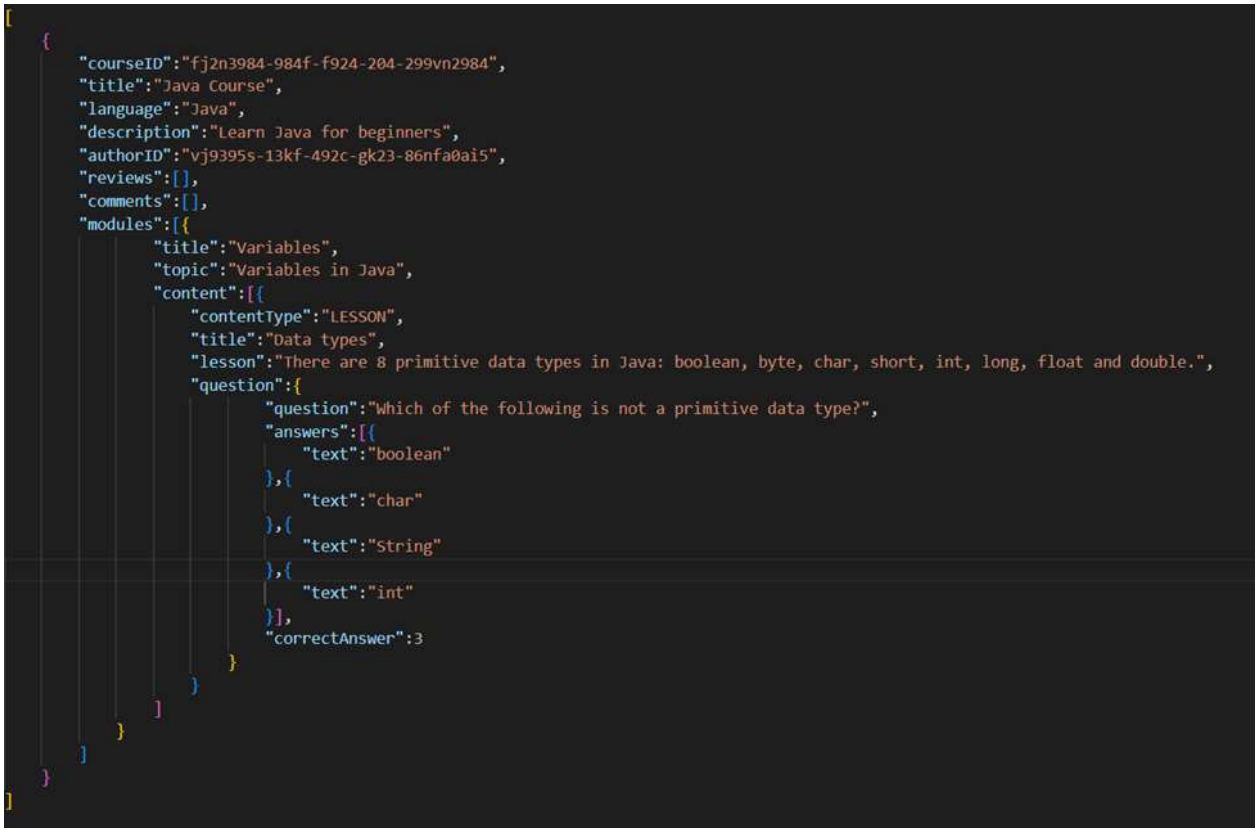
We will store data for users and courses in two files, users.json, and courses.json.

This is an example of how the users.json file would look:

```
[
    {
        "userID":"df6fef1-13kf-492c-gk23-86nfa0ai5",
        "accountType":"STUDENT",
        "firstName":"Bob",
        "lastName":"Smith",
        "username":"bsmith1304",
        "email":"bobsmith@gmail.com",
        "password":"password123",
        "dob":"2/27/1980",
        "enrolledCourses": [{
            "courseID":"256ftw1-15kd-412c-tt13-ka71nvqt8",
            "moduleProgess":[{
                        "progress":[{
                                "complete":1
                            },{
                                "complete":0
                            }
                        ]
                    },{
                        "progress":[{
                                "complete":0
                            },{
                                "complete":1
                            },{
                                "complete":1
                            }
                        ]
                    }
                ]
            }
        ],
        "createdCourses":[]
    },
    {
        "userID":"vj9395s-13kf-492c-gk23-86nfa0ai5",
        "accountType":"AUTHOR",
        "firstName":"Matt",
        "lastName":"Johnson",
        "username":"mjohnson12",
        "email":"mattjohnson@gmail.com",
        "password":"matt$$123",
        "dob":"5/12/1960",
        "enrolledCourses": [],
        "createdCourses":[{
            "courseID":"fj2n3984-984f-f924-204-299vn2984"
            },{
            "courseID":"5oigj-v082i-f024-492-iuqit98"
            }
        ]
    }
]
```

In this example we have the data of two users, one is a student user and the other is an author user. They both have the basic information such as username and password as well their own unique userIDs. They also both have an array of enrolled courses and of created courses. The enrolled courses array is empty for the author user which means they haven't enrolled in any courses yet. The created courses array is empty for the student user because student users are not allowed to create courses. Each enrolled course is represented by the ID of the course as well as the module progress. The module progress is the user's progress through each module in the course. In this example, the course the user has enrolled in has two modules. In the first module, there are two pieces of content. The user has only completed the first content (since the value of "complete" is 1/true) and has completed 2/3 of the contents in the second module. As you can see in the author user, the created courses are just represented as a course ID.

This is an example of how the courses.json file will look:



```
[
    {
        "courseID":"fj2n3984-984f-f924-204-299vn2984",
        "title":"Java Course",
        "language":"Java",
        "description":"Learn Java for beginners",
        "authorID":"vj9395s-13kf-492c-gk23-86nfa0ai5",
        "reviews":[],
        "comments":[],
        "modules":[{
                "title":"Variables",
                "topic":"Variables in Java",
                "content":[{
                    "contentType":"LESSON",
                    "title":"Data types",
                    "lesson":"There are 8 primitive data types in Java: boolean, byte, char, short, int, long, float and double.",
                    "question":{
                            "question":"Which of the following is not a primitive data type?",
                            "answers":[{
                                "text":"boolean"
                            },{
                                "text":"char"
                            },{
                                "text":"String"
                            },{
                                "text":"int"
                            }],
                            "correctAnswer":3
                        }
                    }
                ]
            }
        ]
    }
]
```

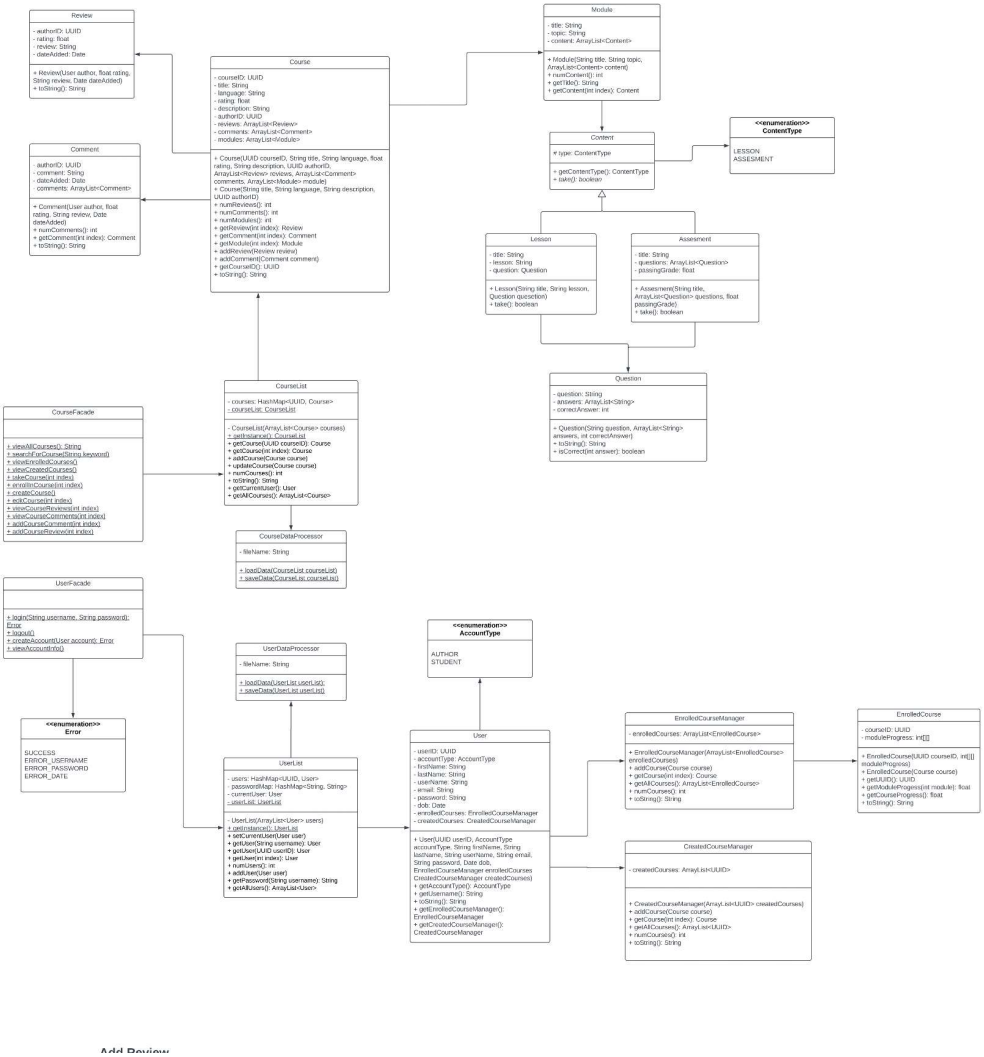

As you can see, each course will have its own unique ID as well as the ID of the user who created the course. Based on the authorID we can see that the author user "mjohnson12" from the users.json example is the author of this course. The course in this example only has one module which is titled "Variables" and the module in this course only has one content. This content is a lesson with the title "Data types". We can see that the lesson is giving information about primitive data types in Java and has a question with 4 different answer. The correctAnswer value is 3 which means "String" is the correct answer.

The course in this example does not have any reviews or comments so here are examples of how each of these would be structured:

Each review has the ID of the user who wrote it, the rating they gave it, the review they gave it, and the date it was added. As you can see by the authorID, the student user from the users.json example is the one who wrote this review. They gave it a 0/10 and said the course was bad.

Each comment has the ID of the user who wrote it, the comment they wrote, the date it was commented, and all the replies to the comment which follow the same structure. As you can see based on the authorID, the student user from the users.json example wrote this comment asking “What is a string?”. Their comment got 1 reply and based of the authorID we can see that the author user from the users.json is the one who wrote the reply. This reply also has 1 reply which is the first user saying thank you.

Class Diagram – Structural Design

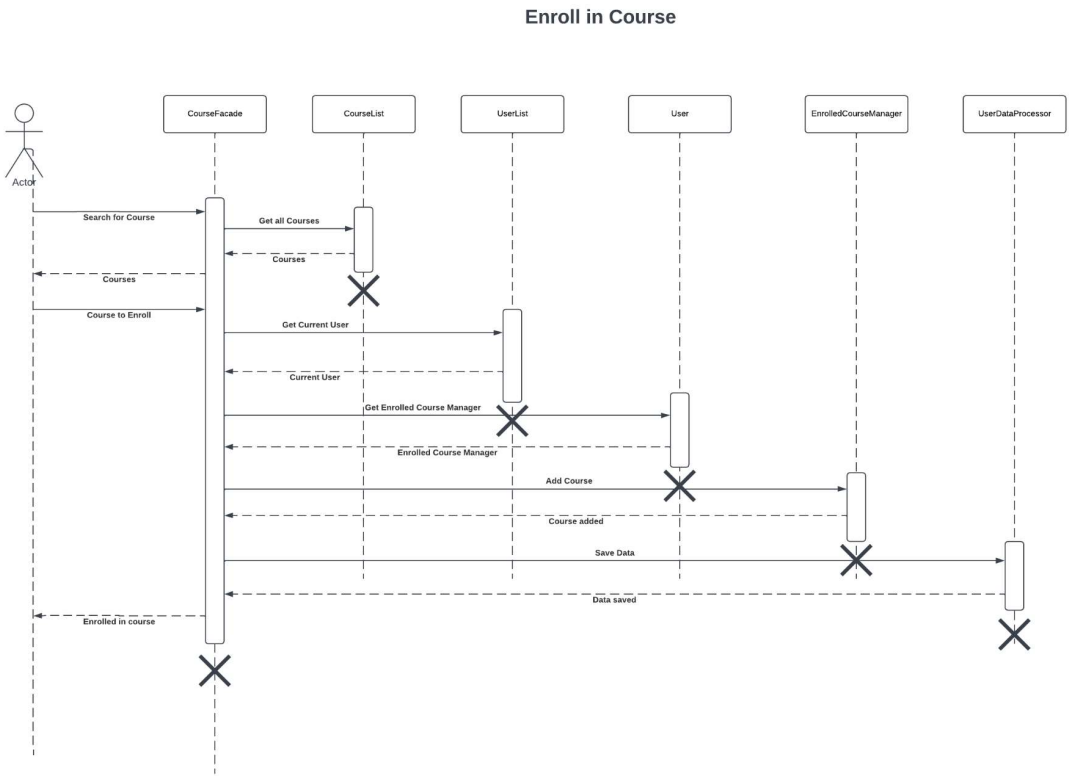


Sequence Diagrams - Dynamic Model

Scenarios

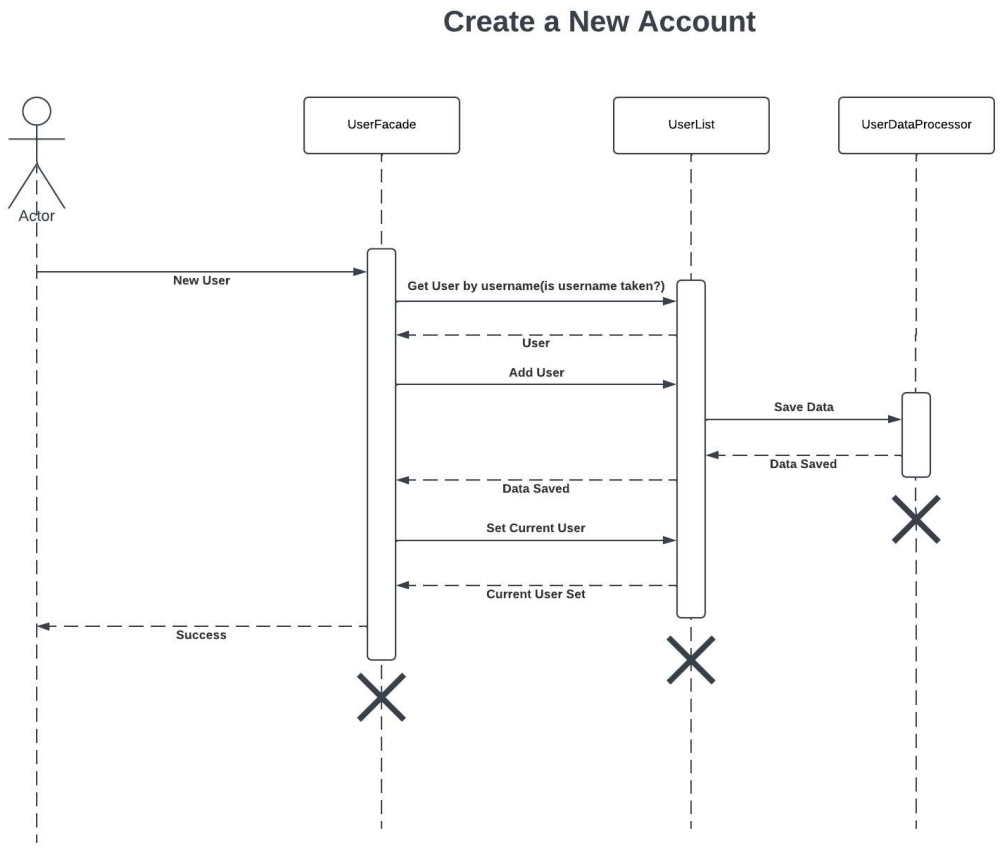
#1 Enroll in course:

This scenario is when the user will want to enroll in a new course. The user will have to first search for their course. Then the course will be added to the users profile as an enrolled course and the data will be saved to the users.json file using the UserDataProcessor class.



#2 Create a new account:

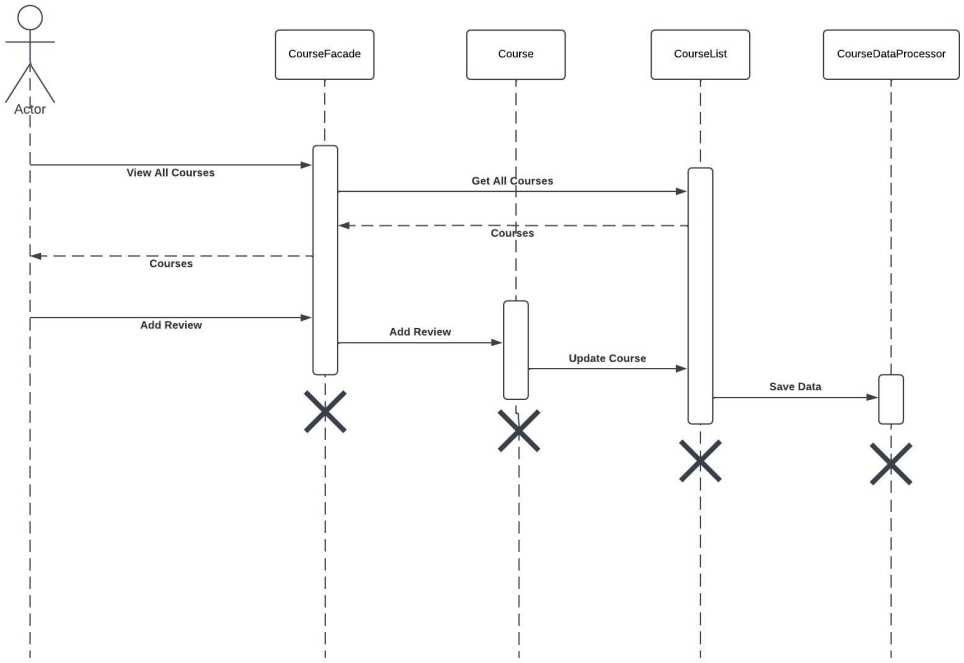
This will be the first thing a new user has to do. They will give all their personal information to the system and then a new user will be created and added to the users.json file.



#3 Add a review

This scenario is for when the user wants to review a course. The user will first view all the courses so they can choose which one to add a review to. Next they will give their review to the system and it will be added to the course and the data will be saved in courses.json.

Add Review


Actor
CourseFacade
Course
CourseList
CourseDataProcessor
View All Courses
Get All Courses
Courses
Courses
Add Review
Add Review
Update Course
Save Data