Design Document

System Overview

ProLang Learning & Applications (name tentative) is a project that aims to provide a website with information and ways to provide an accessible site with multiple languages clients can learn and practice with. Anyone who wants to use our website to learn will be able to. There will also be some select users that can create courses and publish them to the public. The website will contain resources and tools to provide for clients who shall use it to learn and practice their coding skills and have it be accredited to their account. It will provide information and learning tools such as articles, example questions, and various information for multiple languages that are valued within coding that were uploaded by other users. The goal of this product is to create a website that will give an effective understanding of the coding language that the user wants to learn and give resources to do so. The product will be able to provide it in a simple and efficient way. With the rise of a need in coding knowledge and efficiency, businesses and those looking for work, having a website that can help provide tools and resources to help learn is a resource and investment worthy product on this new and innovative product.
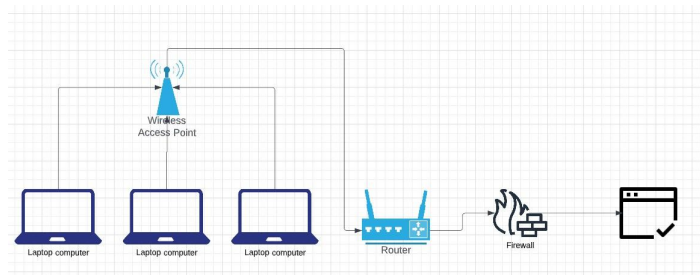
References

Codeacademy: https://www.codeacademy.com/

Coursera: https://ww.coursera.org/

Kenzie Academy: https://go.kenzie.academy/software/engineering/

Environment Overview

This program will be used as a web application. The users will create an account that they can use to save their progress. They will be able to take lessons on any of the available courses on our website. Course makers will be able to create courses and publish them for other users to use and enjoy.

User Interface

Initial Display:

```
Welcome to ProLang

********** Main Menu **********
1. Find a Course
2. Continue a Course
3. Rate a Course
4. Logout

What would you like to do?:
```

Find Course:

```
What would you like to do?: 1

---------- Searching Courses ----------
1. Java
2. Python
3. HTML
Which course would you like to take?: 1
```

Then you will be sent into the start of the Java course.

Continue Course:

```
What would you like to do?: 2

---------- Searching Courses ----------
1. Java
2. Python

Which course would you like to continue?: 1
```

Then you will be sent into the Java course where you left off last.

Course Rate:

```
What would you like to do?: 3

--------- Searching Courses -----------
1. Java
2. Python

Which course would you like to rate?: 1
```

Then you will be sent into the Java Course rating page.

Logout:

```
What would you like to do?: 4

--------- Logging Out ----------
Good bye
```

Data Storage

The data will be stored in three json files: course.json, courseTaken.json and user.json. Each json file holds an array of objects, and each object has a unique identifier. The courseTaken table is the connection between courses and users. It shows which user has started which course and at what date. The diagram below shows the three files are connected. You will see a blown up version of each file afterwards.

```json
[
    {
        "id": 1,
        "courseName": "Java",
        "rating": 4.6,
        "position": 0,
        "courseMaker": "Joe Smith"
    }

    {
        "id": 2,
        "courseName": "Python",
        "rating": 4.5,
        "position": 2,
        "courseMaker": "Sammy Moe"
    }
]
```

```json
[
    {
        "courseID": 1,
        "userID": "2",
        "startDate": "09/01/2019"
    }
]
```

```json
[
    {
        "id": 1,
        "userName": "John Doe",
        "email": "john@gmail.com",
        "DOB": "10/21/1996",
        "password": "johnisepic123"
    }

    {
        "id": 2,
        "userName": "Patrick Brooke",
        "email": "pattyMan@gmail.com",
        "DOB": "08/05/2004",
        "password": "eaglesontop20"
    }
]
```

```json
[
    {
        "id": 1,
        "courseName": "Java",
        "rating": 4.6,
        "position": 0,
        "courseMaker": "Joe Smith"
    }

    {
        "id": 2,
        "courseName": "Python",
        "rating": 4.5,
        "position": 2,
        "courseMaker": "Sammy Moe"
    }
]
```

```json
[
    {
        "courseID": 1,
        "userID": "2",
        "startDate": "09/01/2019"
    }
]
```

```json
[
    {
        "id": 1,
        "userName": "John Doe",
        "email": "john@gmail.com",
        "DOB": "10/21/1996",
        "password": "johnisepic123"
    }

    {
        "id": 2,
        "userName": "Patrick Brooke",
        "email": "pattyMan@gmail.com",
        "DOB": "08/05/2004",
        "password": "eaglesontop20"
    }
]
```
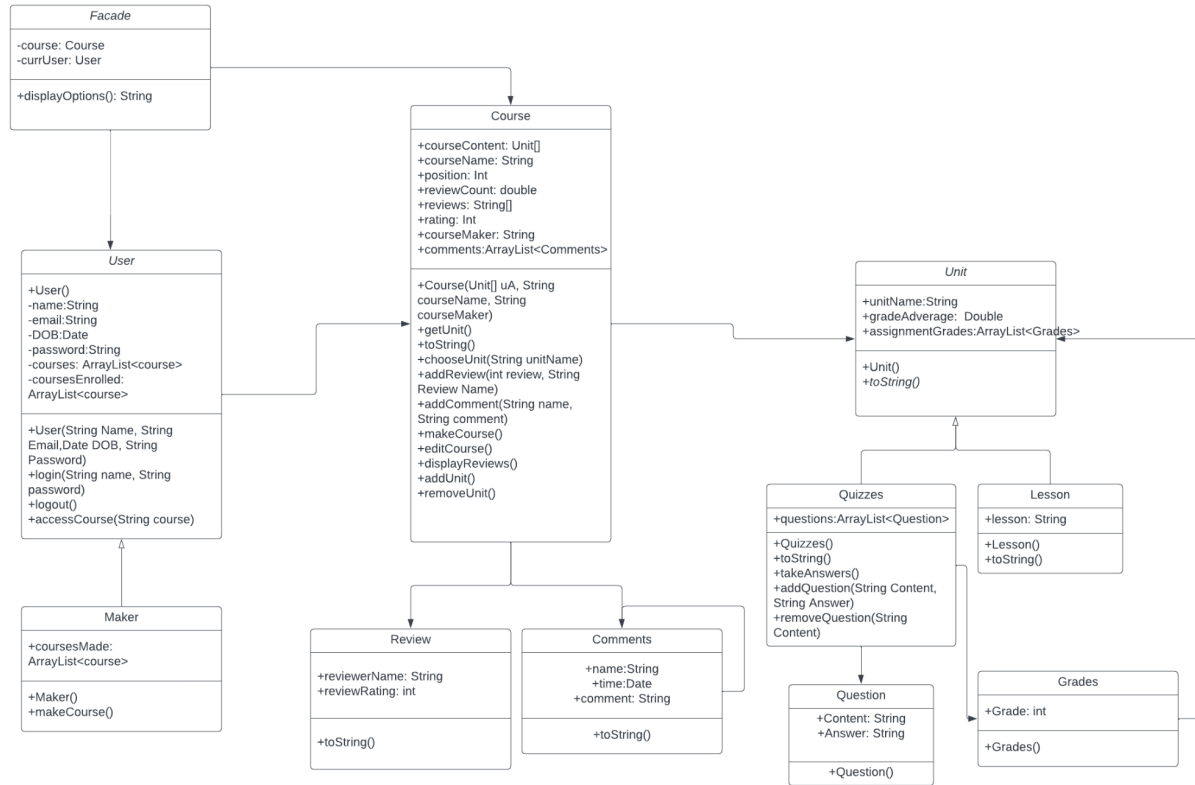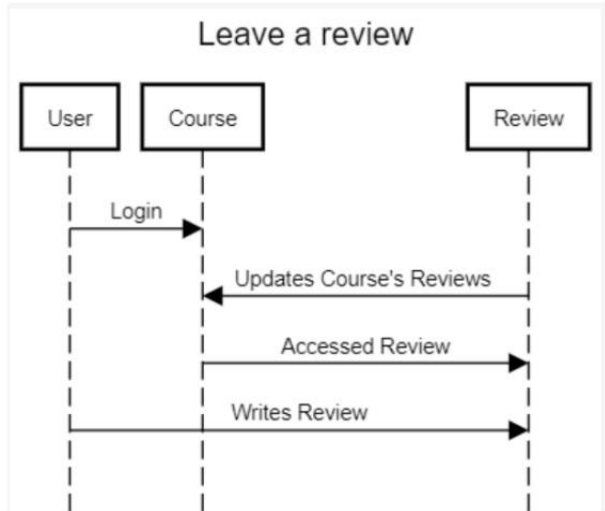
Class Diagram

Facade

-course: Course
-currUser: User

+displayOptions(): String

Course

+courseContent: Unit[]
+courseName: String
+position: Int
+reviewCount: double
+reviews: String[]
+rating: Int
+courseMaker: String
+comments:ArrayList<Comments>

+Course(Unit[] uA, String courseName, String courseMaker)
+getUnit()
+toString()
+chooseUnit(String unitName)
+addReview(int review, String Review Name)
+addComment(String name, String comment)
+makeCourse()
+editCourse()
+displayReviews()
+addUnit()
+removeUnit()

User

+User()
-name:String
-email:String
-DOB:Date
-password:String
-courses: ArrayList<course>
-coursesEnrolled:
ArrayList<course>

+User(String Name, String Email,Date DOB, String Password)
+login(String name, String password)
+logout()
+accessCourse(String course)

Maker

+coursesMade:
ArrayList<course>

+Maker()
+makeCourse()

Unit

+unitName:String
+gradeAdverage: Double
+assignmentGrades:ArrayList<Grades>

+Unit()
+toString()

Quizzes

+questions:ArrayList<Question>

+Quizzes()
+toString()
+takeAnswers()
+addQuestion(String Content, String Answer)
+removeQuestion(String Content)

Lesson

+lesson: String

+Lesson()
+toString()

Review

+reviewerName: String
+reviewRating: int

+toString()

Comments

+name:String
+time:Date
+comment: String

+toString()

Question

+Content: String
+Answer: String

+Question()

Grades

+Grade: int

+Grades()

Sequence Diagrams

Name: User is leaving a review on a course

Description: A user wants to leave a review after taking a course to tell others how great of a course it was. The user will login, go to the course that they want to leave a review for, and update a review.

Leave a review

User	Course		Review

- Login (User → Course)
- Updates Course's Reviews (Review → Course)
- Accessed Review (Course → Review)
- Writes Review (User → Review)

Name: Creating a Course

Description: A course maker wants to create a course to help other people learn and understand a language. The user will login as a maker, and create a course. This will then be published to the website to be used by other users.

Create a Course

Maker	User	Course

- login (Maker → User)
- Maker (User → Maker)
- Make Course (Maker → Course)