Discussion #2

Part 1:

1. The target audience for the translator would be students studying foreign language. Or anyone that would like to learn minimal foreign language without instruction.
2. -The teacher, I want clear instructions of what language is getting translated too for students.

-The student, I want to see correct language translation.

-The user, clear and easy to learn interface without the need for much technical background.

Part 2:

1. The quality attribute I care most about is Reliability because it’s an app that requires accuracy in translating text to a different language. I don’t want a new student memorizing and learning inaccurate material.
2. The attribute for the team to care most about in my opinion is interoperability because it our software doesn’t integrate well with others then it can cause a total crash of any of the apps.
3. -The app must properly scrape accurate text and display proper translation including syntax for less confusion.

-The used data between apps must be consistent for less confusion and proper communication.

1. As a team we choose:

Interoperability: How well can the software integrate with other software?

Reliability: How often does function X succeed?

Part 3:

1. A use case is used for more specific functionality of software to describe the requirements of each interaction the user will come across during operation. The user story is one specific functionality without much detail.
2. Use Case:

Name: guess the translation of the text.

Actor: student

Flow:

1. Student loads the web pages.
2. Student reads instructions for translator and the language of translation.
3. Student enters guess for translation.
4. Student clicks “enter.”
5. Software displays user input in red/green.
6. Software displays correct translated text.
7. Software ask’s user if they would like to play another round.
8. Software displays box window with yes or cancel.

Part 4:

1. Using Planguage+EARS focuses on not only what the system must do but also the known limitation or constraint on the resources and the design. It also wants to know how well the system will do what it accomplishes its task. So Planguage+EARS wants to know functional and non-functional requirements.
2. -Name: language translator guessing game.

-Requirements: When the user enters a translation and clicks the submit button the application shall display the correct translation.

-Rational: The task performs the sole purpose of the application.

-Priority: High.

-Priority Reason: Without proper implementation the application will waste users time and give inaccurate results.

-Status: Planning

-Contact: William Endicott

-Source: William Endicott

-Created By: William Endicott

Version 1, Modified Date 4/15/2021

Part 6:

<https://blog.logrocket.com/methods-for-microservice-communication/>

<https://medium.com/swlh/building-javascript-microservices-with-node-js-d88bf0bb2b92>

<https://www.nginx.com/blog/building-microservices-using-an-api-gateway/>

1. HTTP communication via synchronous or asynchronous communication are two methods we can use. Also, message communication which doesn’t require the need for services to call each other directly and can sometimes eliminate the difficultly of HTTP communication.
2. -HTTP protocol allows for maintainability so one feature maybe down, but the rest of the services are available. With that comes robustness with complete operation with the failure of one functionality. Two drawbacks can be the uniformity factor because then every service might need to use the same interfaces and API endpoints to use the data. Another drawback is the overall difficultly for the level of web app we are making.

-API gateway the gateway provides the API tailored to each client. The API gateway can manage communication with HTTP protocols as well as other unfriendly protocols. Drawbacks include forcing other developers to wait in line for updated gateway. And the need for the updated gateway for each microservice to use the API.

1. The team has decided using web apps and the use of HTTP with be the method we use.