**3.2 Non-Functional requirements**

**3.2.1 Product Requirements**

**3.2.1.1 User Interface requirements**

* The system will be a web-based application; It can be accessed by the following browsers: Chrome, Mozilla Firefox, Safari, Microsoft Edge.
* The system should have 3 main sections where each section is dedicated to a specific type of users.
* The system’s main sections are: Student section, Teacher section , Manager section
* The user interface will be grouped into 3 main sections related with the role of the user.

Log in Interface

* Before the users log in to interact with the system there will be a navigation bar where users can choose how they want to log in. After they choose ,they will have the log in interface .
* It is accessible to all users. It has the input fields for the necessary details, such as the username and password. The submit button will enable users to log in and access the other pages.

Student Interface

* To the students, firstly there will be displayed the home page which includes five categories that make the link to other pages with different functionalities.
* Each of the categories will be displayed using icons and label to be easily determined by the students. On the top there will be a navigation bar for quick navigation through the app.
* In the GetInfo category, there will be displayed 5 options: Check Language, Check Schedule, Check Prices, CourseProgress(The student will be able to see the progress of the chosen courses, to see for example number of topics and chapters completed as well as any evaluation received on part of the teachers),Notification option (The student will get a notification as soon as there is a new post such as: a class is cancelled, the class is postponed, a substitute lesson will be held, etc
* In the Registration category, there will be displayed a form with all the required input boxes, there will be the option to tell the students that the payment has been recorded and a button to make the submission of the form.
* In the Learn category, there will be a chat icon to send the user to a chat section to communicate and to be able to enter a chat with the teacher or with classmates to discuss about the lesson, assignments or questions that arised during the lesson, also the student can call in the course center by clicking the phone icon, or can write messages in the chat boxes, also there will be the camera icon to join in online meetings.
* In the Dairy page there will be two options, MyNotes where the student will be provided with empty pages to write notes for the taken online lessons and to write their review about the center and MySuggestions where the student will be able to provide the center with suggestions.

Teacher Interface

* In the teacher section, there will be displayed all the needed buttons to make the link with other pages. One page will show the information for students where the filtering options will make the searching quicker and easier.
* The first section will be ConductLesson scheduler with features such as: Notes, OnlineLesson, Post, StudentProgress, Attendance
* In the second section there should also be a calendar to show all the dates marked with color in case the lesson is canceled and to keep the participation of the students.
* The second page will show the Notification section in which will be displayed a number of options, they will be notified if schedule change or any other change that could happen ,while this information will be published by manager.
* In third section there will be MyInfo scheduler with features such as MyAccess in their personal data regarding their monthly performance and will have empty pages to keep notes and entering the working hours and Apply where teachers can apply in the beginning of academic year to take a certain course, there will be a menu with all the languages that the center offers to help them to choose the language they will perform.
* In the fourth section will be Communicate (where teacher communicate with manager,make suggestions etc.

Manager Interface

* The manager of the system will have access to all the dedicated interfaces and also, he/she can make limited customizations to the necessary features.
* The system administrator is the only user who can give access permissions to all the other users, so the only way of accessing the system is through a log in where it will require the email and password.
* -A similar design will be used for the prospect, customer, and account creation interfaces. The designated user will add the information about these entities in the field forms, where each field will be grouped according to how relevant or connected it is to the others.

**3.2.1.2 Usability**

The software should have an intuitive user interface that is simple to use and navigate, as well as be user-friendly.

**Accessibility**

* The system can only be accessed if the user has a reliable internet connection and the system administrator has set up an account for them.
* Aesthetics: The application should have a visually appealing and professional design that reflects the brand and mission of the language center.
* Interactivity: The application should be interactive and engaging, with features such as animations, videos, and interactive quizzes to enhance the learning experience.
* Consistency: The application should have a consistent look and feel throughout, with consistent branding, typography, and color schemes.
* Scalability: The system should be designed to scale up or down to accommodate changes in demand, such as seasonal fluctuations or sudden spikes in traffic.
* Maintainability: The system must be easy to maintain and update, with clear documentation and support resources for developers and system administrators.

**Responsiveness**

The app's interface and backend database are engineered for superior responsiveness in addressing user needs. Prompt feedback from design and data transactions is imperative. Moreover, given the likelihood of users utilizing different platforms to access their profile or features, such scenarios must be accounted for. The architecture of the system will be intricate in terms of service uptime. The primary goal of this architecture is to build comparable clones of the system settings so that, in the event that one copy goes down, a load balancer can send traffic to the remaining replica using the same port and IP address.

**Flexibility**

* The system will be powerfully adaptable, which implies that it'll automatically distribute memory and assets to bolster these demands based on the sum of demands.
* Effectiveness
* -The system has almost every feature automated, thus leading to less human errors while having its ability to simplify and streamline the administrative processes of the language school, while also facilitating communication and collaboration between staff, students, and management. By doing so, the app can help the center to run more efficiently, effectively, and smoothly.
* Training recordings on how to utilize the framework appropriately will be given to the desired clients based on their get to on the framework.
* For each startling behavior of the framework there will be simple to get it and self-explanatory blunder messages.

**Efficiency**

• Nearly everyone can complete the majority of the system's operations with the help of drag and drop features, leading to a more productive workflow.

• In order to provide clients a little advantage in learning the route layout, the navigation interface between segments will be based on the most well-known computer application.

• The interface with the highlights will be simple and require fewer steps to operate.

**3.2.1.3 Efficiency**

**3.2.1.3.1 Performance requirements**

* The program will be web-based for administrators and based on an app for users.
* All users who have accounts should be supported by the software program at any time.
* The membership level that the business chooses will have a significant impact on how the computer program runs since greater asset requirements result in higher operating costs. The organize association is another requirement for the framework's operation because the client's organize association's transmission capacity must match the server's transfer speed.
* May computing will ensure the system's continued dependability in operation. In this case, a benefit will have default settings that will determine if the execution is pleasant or not and will distribute more assets as needed to maintain the execution metrics.
* -The capacity of the information is related to the same idea. Depending on the membership, the framework will ask the cloud capacity provider for extra capacity if the volume of information is growing.
* The framework is often server-side-based, which means that the majority of the forms are processed by the server or maybe the client's device, thus clients don't need capable devices to use the framework.

**3.1.2.3.2 Space requirements**

* Scalability**:** Maximum 500 users with the most expensive membership level are predicted to be able to use the framework concurrently without any performance degradation.
* Data storage: The app should have adequate storage space to store data such as user profiles, schedules, class materials, and performance metrics. The app should also provide a backup mechanism to ensure data integrity and prevent data loss.
* Recommended device requirements:
* For PC and laptops:
* CPU  2GHz Dual Core
* User device memory: 4GB RAM, 100 GB HDD or SSD
* Ethernet connection speed: 30 Mbps

**3.2.1.4 Dependability Requirements**

**Availability**

Regarding accessibility, the system must be available around-the-clock without experiencing any performance concerns during business hours. As long as there is an available online connection, it should also be accessible from any location and on any device with a web browser.

**Reliability**

The system will have an internet-based benefit that will allow it to control the essential resources so they work according to plan.

**Monitoring**

Through the cloud computing platform, the framework will be examined for each GET/POST ask and will provide crucial information about the ask's objective, its return value, and any irregularities in the ask body.

The stage of managing the system will monitor and perform the necessary actions with regard to the stack adjustment of the system as well as conducting routine upgrades without compromising the benefit.

**Maintenance**

A self-reporting feature built into the system will let users alert the development team to any bugs they may encounter. This will contain details about the question's subject, the client's actions or events that led to the bug's usage, and other time- and performance-related information.

**Integrity**

Every credential created by the framework administrator will be located in a secure environment that neither clients nor designers may access.

Each segment's director has the ability to add unneeded lower-level customers to the framework according to the privileges that he or she possesses, just as if the client were located in the same office.

Each user must provide their unique information, such as their email address and password, in order to encourage access to the features of the system and their profile.

**3.2.1.5 Security Requirements**

* A only once get-to token will be delivered to the user's email when a modern client is registered by the admin. The customer will be directed to input their secret word and activate the two-factor verification technique using this token.
* The database may be accessed virtually by the system administrator, and clients are provided with basic database access privileges.
* The administrator of the framework would have the option to only allow certain devices to access it, thereby preventing any requests originating from unidentified devices outside of the business.
* The specialists will not compile the unique information of each customer, employee, prospect, client, and account for any purpose. As if a problem were to be discovered, information might be provided to the framework's improvement team for assistance with modifications.