d dddd

HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY OF INFORMATION TECHNOLOGY

GRADUATE THESIS

BUILD AN ENGLISH LEARING WEBSITE

STUDENT NAME:	STUDENT ID:
NGUYỄN THANH LÂM	17110324
School year:	2017 – 2021
Major:	INFORMATION TECHNOLOGY
SUPERVISOR:	Dr. LÊ VĨNH THỊNH

Ho Chi Minh, June 2020

HO CHI MINH UNIVERSITY OF TECHNOLOGY AND EDUCATION
FACULTY OF INFORMATION TECHNOLOGY

GRADUATE THESIS

BUILD AN ENGLISH LEARING WEBSITE

STUDENT NAME:	**STUDENT ID:**
NGUYỄN THANH LÂM	**17110324**
School year:	**2017 – 2021**
Major:	**INFORMATION TECHNOLOGY**
SUPERVISOR:	**Dr LÊ VĨNH THỊNH**

Ho Chi Minh, June 2020

COMMENTARY OF SUPERVISOR

Student name: Nguyễn Thanh Lâm Student ID: 17110324

Major: Information Technology.

Topic name: Build an English learning website.

Name of supervisor: Dr. Lê Vĩnh Thịnh.

COMMENTARY

1. **On content of topic & workload done:**

....

....

....

2. **Advantage:**

....

....

....

3. **Disadvantage:**

....

....

....

4. **Recommend for defense or not?...**
5. **Rating type: ....**
6. **Mark: ....(By word:..)**

Ho Chi Minh, ……………………………. 2020

SUPERVISOR

(Name and signature)

COMMENTARY OF REVIEWER

Student name: Nguyễn Thanh Lâm Student ID: 17110324

Major: Information Technology.

Topic name: Build an English learning website.

Name of reviewer: Dr. Huỳnh Xuân Phụng.

COMMENTARY

7. **On content of topic & workload done:**

....

....

....

8. **Advantage:**

....

....

....

9. **Disadvantage:**

....

....

....

10. **Recommend for defense or not?...**

11. **Rating type: ....**

12. **Mark: ....(By word: ....)**

Ho Chi Minh, ………………………………. 2020

REVIEWER

(Name and signature)

ASSURANCE

I guarantee that this project is my work. I do not copy or use any documents or source code of others without specifying the source. I take responsibility for any violation.

Ho Chi Minh, January 13th 2020

Lâm

Nguyễn Thanh Lâm.

MANY THANKS

In fact, no success is not tied to the direct or indirect support or help of others. Now, I would like to send my most sincere thanks to Dr. Le Vinh Thinh, who supported and motivated me in the process of choosing topics, guidance, and suggestions. Although I do not ask much when asking for help, it is my motivation to complete the project. Without the guidance and practical experience of the teacher, I think this project will be difficult to complete and complete on time. Once again, I would like to thank my supervisor.

I would also like to thank Dr. Le Vinh Thinh for directly guiding me to complete this project. I wouldn't be able to finish without your guidance. Due to the lack of experience in designing and building software applying this relatively new process, there may be many errors in the construction process, please sympathize.

Due to limited time, limited knowledge, and many other errors, mistakes will inevitably be made, we hope to receive valuable comments from everyone for further improvement in the future. Thank you sincerely.

Ho Chi Minh, January 13th 2020

Lâm

Nguyễn Thanh Lâm.

KẾT QUẢ ĐẠT ĐƯỢC CHO ĐẾN HIỆN TẠI (NGÀY 13/01/2020)

STT	TÊN CÔNG VIỆC	TÌNH TRẠNG	NGƯỜI PHÁT TRIỂN
1	Chức năng Đăng Ký	Hoàn thành	Nguyễn Thanh Lâm
2	Chức năng Đăng Nhập Bằng Tài Khoản	Hoàn thành	Nguyễn Thanh Lâm
3	Chức năng Đăng Nhập Bằng Tài Khoản Google	Hoàn thành	Nguyễn Thanh Lâm
4	Chức năng Đăng Ký	Hoàn thành	Nguyễn Thanh Lâm
5	Chức năng Xem Thông Tin Tài Khoản	Hoàn thành	Nguyễn Thanh Lâm
6	Chức năng Chỉnh Sửa Thông Tin Tài Khoản	Hoàn thành	Nguyễn Thanh Lâm
7	Chức năng Học Từ Vựng	Hoàn thành	Nguyễn Thanh Lâm
8	Chức năng Thêm Mới Từ Vựng	Hoàn thành	Nguyễn Thanh Lâm
9	Chức năng Chỉnh Sửa Từ Vựng	Hoàn thành	Nguyễn Thanh Lâm
10	Chức năng Xóa Từ Vựng	Hoàn thành	Nguyễn Thanh Lâm
11	Chức năng Học Ngữ Pháp	- Đã Hoàn Thành Phần Back-End(API). - Đang Hoàn Thành Phần Xử Lý và Giao Diện.	Nguyễn Thanh Lâm
12	Chức năng CRUD Ngữ Pháp	- Đã Hoàn Thành Phần Back-End(API). - Đang Hoàn Thành Phần Xử Lý và Giao Diện.	Nguyễn Thanh Lâm
13	Chức năng Luyện Tập Từ Vựng và Ngữ Pháp	- Đã Hoàn Thành Phần Back-End(API).	Nguyễn Thanh Lâm

		- Đang Hoàn Thành Phần Xử Lý và Giao Diện.	
14	Chức năng CRUD Luyện Tập Từ Vựng và Ngữ Pháp	- Đã Hoàn Thành Phần Back-End(API). - Đang Hoàn Thành Phần Xử Lý và Giao Diện.	Nguyễn Thanh Lâm
15	Chức năng Thi Thử Toeic	- Đã Hoàn Thành Phần Back-End(API). - Đang Hoàn Thành Phần Xử Lý và Giao Diện.	Nguyễn Thanh Lâm
16	Chức năng CRUD Thi Thử Toeic	- Đã Hoàn Thành Phần Back-End(API). - Đang Hoàn Thành Phần Xử Lý và Giao Diện.	Nguyễn Thanh Lâm
17	Chức năng Quản Lý Thông Tin Tài Khoản Người Dùng	- Đã Hoàn Thành Phần Back-End(API). - Đang Hoàn Thành Phần Xử Lý và Giao Diện.	Nguyễn Thanh Lâm
18	Chức năng Quản Lý Quyền Hạn Tài Khoản Người Dùng	- Đã Hoàn Thành Phần Back-End(API). - Đang Hoàn Thành Phần Xử Lý và Giao Diện.	Nguyễn Thanh Lâm

TABLE OF CONTENT

Contents

CHAPTER 1. OVEWVIEW

1.1. Urgency and objectives of project

1.1.1. The urgency of project

In recent years, the investment boom of foreign companies in Vietnam has created more and more job opportunities for graduates, especially students of engineering disciplines. Therefore, the need to communicate in English at work has increased significantly.

In particular, it is difficult to escape the dominance of English in our global world. Although there are only an estimated 360 million native speakers, a total of 1.6 billion people speak English as a first, second, or third language. Over the centuries, English has become the "language of the bridge" that develops with global trade, diplomacy, and culture.

Knowing foreign languages is not only an indispensable requirement of highly skilled workers to meet the constantly renewed technological processes, but also knowing a foreign language is also a necessary capacity for modern Vietnamese. Proficiency in foreign languages is gradually becoming an indispensable skill in the current economic integration context. Foreign language ability is the top standard for large companies to recruit employees as well as promote management positions. Although there are currently no official statistics on the correlation between foreign language proficiency and salary, the reality shows that the window of opportunity is often more open for candidates with more than one language. By understanding foreign languages, people can have a deeper understanding of world civilization, expand cooperation, exchange, and develop their own potentials.

People often say "most English, second computer" has shown the role and importance of foreign languages. Learning a foreign language requires time, perseverance, and application-level of learners in practice.

Stems from the desire to give everyone a source of knowledge about English for everyone to have access to.

From there, it helps people to have knowledge resources to learn and practice by themselves. To exchange English knowledge and skills for yourself. Since then, creating more and more opportunities for myself in work and in life to go further on my career path and have a quality life.

Especially in this era of information technology 4.0. High-speed, fast, and convenient internet access is easy for everyone.

That is why I chose this specialized topic to research.

1.1.2. Project objectives

Topic "Building English learning website" to solve the issues include:

- Using technologies, techniques, application libraries in the software development process, providing technology solutions to solve, optimize and bring customers the best value, including Application technology helps to develop the website, application technologies help to develop project fastest, deploy most independently, and easily to different environments.
- Build an English learning website that allows users to register an account, log in to the website to learn vocabulary, grammar, and practice questions.
- Build an operation management website that allows admin to manage Account, Vocabulary, Grammar, Practice.
- In addition to completing functions for system users, security factors such as server-user authentication, private information encryption algorithm, and decentralization mechanism must be met, mail system, ...
- The product is a website, so it requires a friendly interface, in accordance with the desired topic. The product is also required to be used on many browsers as well as many types of mobile devices and PC to the new feature level can work well, the interface does not have too much variation between device and browser.

1.2. Objects, scope and methods of research

1.2.1. Research objects

Project are conducted around focus subjects, including technologies and knowledge of website management and operation, knowledge, and skills in English.

In which, technology objects include compulsory research objects: open source ASP.NET Core, RESTful APIs, Angular framework, MS SQL Server database. Technologies used to deploy and operate the system include Firebase, Heroku. In addition to incorporating technology research objects, some valuable libraries such as JWT (Json Web Token), socket.io, ngx-chart, ng-bootstrap also need to be studied for application. Use of product functionality.

Study subjects include: operating and managing a website, knowledge, and skills in English.

1.2.2. Research scope

The scope of the research is set at a general level, understands the general knowledge of the research content and applies each knowledge content to the actual product, without putting heavy theories and non-application.

1.2.3. Research methods

Researching and understanding the materials, websites related to the applied technologies.

Consult with your instructor and friend who have experience relating to issues related to the essay topics to create the accuracy of the topic.

Refer to some English websites with many visitors such as Tiếng Anh TFlat, Tiếng Anh Mỗi Ngày, Doulingo,...

1.3. Expected results

With the results of collecting and examining the current systems, the new expected solution help brings together the subsystems to create a complete, unified, and friendliest with users. The system will include the following sections:

- English learning website.
- Website operation management for admin.

The system is built-in RESTful APIs and Microservices architecture deployed and operated on Heroku, utilizing Docker Engine technology to help reduce deployment time and configure services across each environment. In addition to responding to availability and load balancing, the system is always operating continuously despite the incident in each service.

CHAPTER 2. THEORETICAL BASIS

2.1. Overview of ASP.NET Core

2.1.1. Introduction to ASP.NET Core

ASP.NET Core is the new version of the ASP.NET web framework mainly targeted to run on .NET Core platform.

ASP.NET Core is a free, open-source, and cross-platform framework for building cloud-based applications, such as web apps, IoT apps, and mobile backends. It is designed to run on the cloud as well as on-premises.

Same as .NET Core, it was architected modular with minimum overhead, and then other more advanced features can be added as NuGet packages as per application requirement. This results in high performance, require less memory, less deployment size, and easy to maintain.

ASP.NET Core is an open source framework supported by Microsoft and the community, so you can also contribute or download the source code from the ASP.NET Core Repository on GitHub.

2.1.2. Development history [13]

ASP.NET has been used from many years to develop web applications. Since then, the framework went through a steady evolutionary change and finally led us to its most recent descendant ASP.NET Core 1.0.

ASP.NET Core 1.0 is not a continuation of ASP.NET 4.6.

It is a whole new framework, a side-by-side project which happily lives alongside everything else we know.

It is an actual re-write of the current ASP.NET 4.6 framework, but much smaller and a lot more modular.

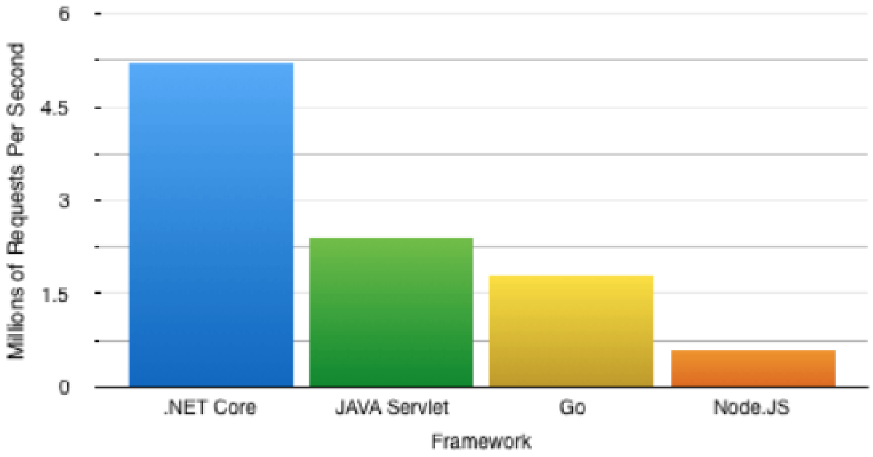
Some people think that many things remain the same, but this is not entirely true. ASP.NET Core 1.0 is a big fundamental change to the ASP.NET landscape.

Table 2.1. Development history of ASP.NET Core

Version Number	Release Date	End of Support	Supported Visual Studio Version(s)
1.0	2016-06-27	2019-06-27	Visual Studio 2015, 2017
1.1	2016-11-18	2019-06-27	Visual Studio 2015, 2017
2.0	2017-08-14	2018-10-01	Visual Studio 2017
2.1	2018-05-30	2021-08-21	Visual Studio 2017
2.2	2018-12-04	2019-12-23	Visual Studio 2017 15.9 and 2019 16.0 preview 1
3.0	2019-09-23	2020-03-03	Visual Studio 2017 and 2019
3.1	2019-12-03		Visual Studio 2019

2.1.3. Features [13]

- **Cross-platform & container support:**
  - ○ Windowns, Linux, macOS.
  - ○ Docker, Kuberenetes.
- **High performance:**
  - ○ The new Kestrel web server was redesigned from the ground up to take advantage of asynchronous programming models, be much more lightweight, and fast!



- **Asynchronous via async/await.**

○ ASP.NET has excellent support for utilizing asynchronous programming patterns.

○ Async is now implemented in all common .NET Framework classes and most third-party libraries. Most modern applications spend most of their time and CPU cycles waiting for database queries, web service calls, and other I/O operations to complete.

- **Unified MVC & Web API frameworks:**

○ With ASP.NET Core, MVC and Web API have been merged together. There was always a lot of overlap with the two frameworks. MVC could always return JSON data instead of HTML. Combining them was a good move and simplifies development.

○ With ASP.NET Core we also have the new Razor Pages. They extend the MVC framework to allow encapsulating the controller and model aspects of a page together with two-way binding. They are sort of a replacement for WebForms while using the familiar Razor syntax.

- **Multiple environments and development mode:**

○ It allows you to easily differentiate parts of your code for their behavior in development, staging, production, etc.

- **Dependency Injection:**

○ One of the great new features of ASP.NET Core is built in dependency injection. It is heavily used within ASP.NET MVC itself.

○ It is the preferred way that things like logging contexts, database contexts, and other things are passed into your MVC controllers.

- **WebSockets & SignalR:**

○ ASP.NET has first class support for WebSockets. This can be used to persist long running connections and communicate back and forth with the browser.

○ SignalR is a full framework that is also available that makes it easy handle common scenarios.

- **Cross-Site Request Forgery (CSRF) Protection.**

○ Security is important. It is also one of those things that can be a lot of work to prevent certain types of attacks. CSRF is in referencing to hijacking users authenticated session to perform an action that they did not initiate.

○ ASP.NET has a good framework that is available to prevent these types of attacks. It generates anti-forgery tokens.

- **"Self hosted" Web Applications:**

○ Sometimes you need to make a web application that will be deployed on to a desktop and not a server running IIS. Our free ASP.NET profiler, Prefix, is a perfect example of this. Its front end is all HTML that is loaded from an ASP.NET application running as a Windows Service.

○ With ASP.NET Core, you can also use the standard Kestrel web server. One of the great advantages of .NET Core is that your web application is essentially a console application. IIS just sits in front of it as a reverse proxy. This means that you can also deploy your app only with kestrel for non-server based use cases, like Prefix.

- **Action Filters:**

○ One of the great features of ASP.NET is the support for extensible filters. This allows you to implement functionality that can be applied to an entire controller or action without modifying the action itself.

○ Filters are used to specify caching, error handling, authorization, or any custom logic you would like to implement.

- **Extensible Output Caching:**
  - This feature allows ASP.NET to cache the output generated by a page and serve this cached content for future requests. It stores the data that is not updated frequently and outputs that specific data from a cached location.
  - ASP.NET makes it easy to specify how long any request should be cached via common HTTP headers. It also has support for caching output within the memory on your web server itself. You can even use Redis or other providers to handle your output caching.
- **Globalization and Localization:**
  - ASP.NET makes it easy to localize dates, numbers, and the text within your web application. If you want your application to be used across the globe, localization will be very important to you.
  - ASP.NET enables customizing your application for multiple languages via resource files. These resource files are considered as the central repository where all texts are kept, and web pages can read this resource file and get labels populated. There are two types of resources:
    - Local Resources – specific for a page (i.e., there will be local resource file for every page).
    - Global Resources – common for the whole website (i.e., one resource file accessed by all pages).
- **Swagger OpenAPI:**
  - If you are creating API applications, you want to make sure you are using Swagger. It makes it easy to document and test your APIs.
  - ASP.NET has historically provided built-in functionality that is pretty similar for SOAP web services created with WCF. If you are using Web API or MVC for RESTful APIs, you definitely want to use Swagger.

2.1.4. Introduction to Web APIs in ASP.NET Core [9]

ASP.NET makes it easy to build services that reach a broad range of clients, including browsers and mobile devices. With ASP.NET you use the same framework and patterns to build both web pages and services, side-by-side in the same project.

ASP.NET was designed for modern web experiences. Endpoints automatically serialize your classes to properly formatted JSON out of the box. No special configuration is required. Of course, serialization can be customized for endpoints that have unique requirements.

Secure API endpoints with built-in support for industry standard JSON Web Tokens (JWT). Policy-based authorization gives you the flexibility to define powerful access control rules—all in code.

ASP.NET lets you define routes and verbs inline with your code, using attributes. Data from the request path, query string, and request body are automatically bound to method parameters.

.NET provides first class support for HTTPS out of the box. Automatically generate a test certificate and easily import it to enable local HTTPS so you run, and debug, your apps the way they are intended to be... secured.

Build, debug, and deploy from any platform to any platform.

2.1.5. Advantages of ASP.NET Core [10]

- Build hybrid, Native & Interactive Apps with ASP.NET Core that works on any platform.

- You will get a better response time with high performance for your application.

- You can run codes of various architecture with the help of Cross Architecture Consistency.

- Get the advantage of Framework Dependent and Self-contained deployment.

- You can run multiple versions of an application on the same computer.

- Get rid of command-and-control development with open source.

- Get advantage of the Portable and Runnable Framework.

- Build your applications on an evolved framework.

- Create interactive client-side web UI on C# instead of Java with the help of Blazor.

- Easily extract your data without any coding with the help of Model Binding.

- Create a big application easily with Model-View-Controller.

- You will have the ability to host on Docker, Apache, IIS, and self-hosting.

- You will get outstanding support for using asynchronous programming patterns.

- When security is your biggest concern, .NET Core is the ultimate solution.

- Open-Source - It is an open-source and community-focused web framework.

- You will have the power to control your applications with Middleware.

- You will get effortless updates from Microsoft.

- You will find ASP.NET Core very easy to maintain.

- You will have support for cloud-based developments.

- ASP.NET provides you to Optimize data access.

- A unified story for building web UI and web APIs.

- Support Built-In Dependency Injection - It supports built-in Dependency Injection.

- Supports Modular - It support modular HTTP request.

- Integration of Modern UI Framework - ASP.NET Core support modern, a Client-side framework like AngularJs, ReactJs and React with Redux etc. ASP.NET framework supports client-side framework templates like AngularJs, ReactJs and React with Redux etc.

2.1.6. ASP.NET Core appied in project

- User API Gateway, Admin API Gateway.
- Users API, Vocabularies API, Grammars API, Practices API, QuestionReadings API, Paragraphs API.

2.2. Angular Framework

2.2.1. Introduction to Angular

Angular is a platform and framework for building single-page client applications using HTML and TypeScript.

Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your apps.

The architecture of an Angular application relies on certain fundamental concepts. The basic building blocks are NgModules, which provide a compilation context for components. NgModules collect related code into functional sets; an Angular app is defined by a set of NgModules. An app always has at least a root module that enables bootstrapping, and typically has many more feature modules.

Components define views, which are sets of screen elements that Angular can choose among and modify according to your program logic and data.

Components use services, which provide specific functionality not directly related to views. Service providers can be injected into components as dependencies, making your code modular, reusable, and efficient.

Modules, components and services are classes that use decorators. These decorators mark their type and provide metadata that tells Angular how to use them.

The metadata for a component class associates it with a template that defines a view. A template combines ordinary HTML with Angular directives and binding markup that allow Angular to modify the HTML before rendering it for display.

The metadata for a service class provides the information Angular needs to make it available to components through dependency injection (DI).

An app's components typically define many views, arranged hierarchically. Angular provides the Router service to help you define navigation paths among views. The router provides sophisticated in-browser navigational capabilities.

2.2.2. Development History

Angular (commonly referred to as "Angular 2+" or "Angular v2 and above") is a new release of AngularJS, released on September 14, 2016, was a complete rewrite of the previous one, entirely based upon the new ECMAScript version 6 (officially ECMAScript 2015) specifications. Just like the ASP.NET Core rewrite, the “revolution” brought such a number of breaking changes at the architectural level, HTTP pipeline handling, app lifecycle, state management, that porting the old code to the new one was nearly impossible: despite keeping its former name, the new

Angular version was a brand-new framework with little or nothing in common with the previous one.

The choice of not making Angular 2 backward compatible with AngularJS clearly demonstrated the intention of the author's team to adopt a completely new approach: not only in the code syntax but also in their way of thinking and designing the client app. The new Angular was highly modular, component-based, comes with a new and improved dependency injection model and a whole lot of programming patterns its older cousin never heard of.

Table 2.2. Development history of Angular

Version	Release Date
Angular 2	2016-09-14
Angular 4	2017-03-23
Angular 5	2017-11-01
Angular 6	2018-05-04
Angular 7	2018-10-18
Angular 8	2019-05-28
Angular 9	2020-02-06
Angular 10	2020-06-24

2.2.3. Features

- Cross-Platform:

With Angular, you can develop progressive web applications (PWA). PWA offers an app-like experience to the audience using modern web capabilities. As per your needs, you can deploy an app as native as well as progressive. The hybrid mobile SDK called Ionic can ship your apps to the app store and deploy the same to the mobile web as PWA.

In addition, you can develop apps for the desktop with Angular.

- High Speed & Optimum Performance:

Angular redefines the modern JavaScript virtual machine by turning templates into code. Thus, your hand-written code can leverage on a productive framework. The best thing about Angular is that you can render the code into HTML & CSS and have the first view of the application on any other platform like NodeJS, .Net, PHP, etc.

The loading time of Angular apps is faster than any other front end framework in the industry. It loads at the speed of Cheetah with any new component router. Further, the code gets split automatically as per the user's loading and rendering requirements.

- Angular Applications for Everyone:

Angular is a magical front end platform that not only allows developing mind-blowing applications but also enables to create of high-end animations to enhance the user experience. The API of Angular is so intuitive that developers can brew complex choreographic as well as animation with low code.

In addition, with smart unit testing frameworks like Jasmine and Karma, you can fix your broken code anytime. Angular has more than 11 build-in testing modules to ensure error-free code.

Building accessible applications are one of the breakthrough capabilities of Angular. Everyone including people with special abilities can use accessible applications.

- MVC Architecture:

MVC stands for Model-View-Controller. The application data is managed by the Model and View manages the data display. While the controller plays as a connector between the view and model layers. Generally, in MVC architecture, you can split up the app accordingly and write code to connect the same. However, in Angular; developers just need to split up the application into MVC and it does the rest on its own. And it saves a significant amount of coding time.

- Efficient Two-Way Data Binding:

The most effective Angular feature is its two-way data binding system. The View layer exactly represents the Model layer and they stay in perfect synchronization. If you make any change in the model, the users can see in the view model automatically. Thus, it decreases important development time.

- Less Code Framework:

Angular is a low code framework comparing to the other front end technologies. You don't need to write separate code to link the MVC layers. And you don't require specific code to view manually also. In addition, the directives are isolated from the app code. These all together automatically minimize the development time.

- Angular CLI (Command Line Interface):

The Angular CLI follows the industry-best-practices for frontend development with striking build-in features SCSS support or routing. Moreover, the common Angular CLI like ng-new or ng-add supports the developers to discover ready-made features with ease.

The basic CLI of Angular is:

+ ng New: It's a primary initial for any Angular app development. And you can also create a new workplace with this command.

+ ng Generate: It allows you to create new components, services, routes, and pipes. Also, you can create simple test shells with ng Generate.

+ ng Serve: It permits you to test your Angular app on a local server.

+ Test, Lint: Shine your code with smart unit and end-to-end tests. And ng lint helps to run programs to analyze code for possible errors.

- CDK and Angular Material:

Angular as a top frontend language has been improving its Component Development Kit (CDK) with the version upgrades. Refreshing and virtual scrolling are the prominent features of the latest Angular CDK. It helps in dynamic loading as well as unloading of the DOM to create a large list of high-performing data. You can import the ScrollingModule or DragDropModuleto the application.

Some of the prominent features of this DragDropModule are sorting within a list, free dragging, previews, and placeholders. In addition, it helps in reordering list (moveItemInArray) and transferring items between the lists (transferArrayItem).

- Virtual Scrolling:

Angular Virtual Scrolling helps the code react to various scroll events. Besides loading and unloading visible DOM elements, Virtual Scrolling enables an effective way for items simulation.

- TypeScript:

With TypeScript, Angular offers a seamless experience to the developers. And TypeScript is the top choice for frontend development in 2019. It is highly efficient in detecting bugs and thereby, reduces developing time. TypeScript also automatically populates the root file configuration for easy compilation. In comparison to JavaScript, TypeScript has richer enums, interfaces, generics, hybrid types, union/intersection types, access modifiers, etc.

- Dependency Injection:

The built-in dependency injection of Angular makes application development easier for developers. It just asks your dependencies. Just ask it 'I need y'! And it creates the same & delivers it to you.

- Directives:

Directives are the most challenging Angular features. You can create custom HTML tags serving as custom widgets with the help of directives. Developers can use these to decorate behavior-driven elements. You can also manipulate DOM attributes as per needs with the help of directives.

- Putting Everything at One Place:

Angular is already in a bright place in the market. Everyone is going gaga over this leading framework for its extraordinary accessibility and low code features. Various leading websites like Samsung Forward, Deutsche Bank Developer Portal, Microsoft Office Home, Delta, Forbes, BMW Find a Dealer, BMW Driveaway Price Calculator, Google Marketing Platform, and many more are using Angular in their web development and the result is known to everyone. So, Go Angular!

2.2.4. Advantages of Angular

- Maintained by Google:

The best thing about Angular is that it is developed and maintained by Google. As it is backed by a trusted company, trust is well established within the community. Developers believe in the framework and in the fact that it will be maintained, and issues will be resolved, of course with the help of community contributions.

- Large community and ecosystem:

Angular is actively maintained and has a large community and ecosystem. You can find lots of material on this framework as well as many useful third-party tools. Currently, Angular has around 63K stars on GitHub, a clear indication of the popularity of the framework.

- Two-way Data binding:

Two-way data binding in Angular will help users to exchange data from the component to view and from view to the component. It will help users to establish communication bi-directionally. Two-way data binding can be achieved using a ngModel directive in Angular. This ensures that the model and the view are always kept in sync without any extra effort.

- Use of TypeScript:

Angular opted for TypeScript as the primary programming language. TypeScript adds a lot of syntactic sugar to JavaScript. With TypeScript, we also get several productivity benefits. In editors like VS Code and WebStorm, we have access to robust code help (IntelliSense) as we type, making it easier to discover types, the features they offer, and most importantly the common syntactical errors.

- Support for i18n:

Internationalization is the process of making sure that an application is designed and prepared to be used in regions with different languages. Localization is the process of translating your internationalized app into specific languages for particular locales. Angular can take care of most things when it comes to multiple languages. Dates, numbers, times, and other things are easily taken care of based on the locale. On top of that, the Angular CLI allows us to install the @angular/localize package and generate most of the boilerplate code necessary.

- Support for lazy loading:

An Angular module is a logical unit of code that performs a certain task or takes care of a certain responsibility in the application. A normal Angular application can be broken down into multiple modules, where each module is delegated a certain responsibility. A module can comprise components, pipes, directives, or more modules.

Breaking down an application into modules not only allows you to easily manage an application as it grows but also offers huge performance advantages. Instead of loading all the application code at once in the browser which may cause more loading time and a slower initial render of the web page, it loads only what is necessary. This is called lazy loading.

Only the modules that are required are loaded initially and more modules or rather chunks of code are loaded as when required. This offers seamless performance and smaller loading times.

Every Angular app has at least one NgModule class, the root module which is responsible to bootstrap the project. Conventionally, it is named AppModule, and it resides in a file named app.module.ts.

- Dependency Injection:

Dependency injection or DI is one of the most important application design patterns and Angular has its own dependency injection framework. It is typically used in the design of Angular applications to increase their efficiency and modularity. This is closely related to the previous point - modularization of the Angular app.

Dependencies are services or objects that a class needs to perform its function. The class requests services from an external resource, generally another module or class in this case, instead of creating them themselves. This allows us to delegate a single responsibility to each module.

The advantage of the dependency injection design pattern is to divide the task into multiple services. The class will not create the dependent object itself, rather it will be created and injected by an Angular injector. Dependency Injection is a technique that is used to reduce the coupling amongst various components of an application. Looser coupling allows for better testability and maintainability.

- Powerful Router:

Routing or in-app navigation in Angular is taken care of by the Angular Router. It is a module available in the @angular/router package. The angular router is a very powerful and flexible navigational service. It uses a component called router-outlet to load various components into the view depending upon the URL in the browser.

Angular Router has built-in support for lazy loading components, so the modules are loaded from the server as the user navigates within the app. Huge apps have a lot of benefits because of this feature.

- Support for Ionic:

Angular was the first framework that was supported by the Ionic framework. It allows building cross-platform mobile applications for Android and iOS using the Ionic components and Angular code. As of now, more frameworks like ReactJS and VueJS are also supported but still, Angular is widely used with the Ionic framework for mobile application development.

- Availability of Packages:

There is a huge repository of open-source packages available for Angular developers. Some of the most important packages are NgBootstrap, Angular Google Maps, NgRx, NgTranslate, AngularFire, NgxTextEditor, Angular Material, Ng2 Pdf Viewer, NgxCharts, and many more. Some of these packages are maintained by the official Angular team and most of these are taken care of by the community of developers as well. It is most likely that if you are looking to build a complex feature or a component for your application, a package already exists for you, so why reinvent the wheel?

2.2.5. Disadvantages of Angular

- Steeper learning curve:

While Angular is great, it may not be the case for a complete beginner. Even if you have experience with HTML, CSS, and JS, you may feel a little bit uncomfortable with the steep learning curve that it has. It brings its own notions and philosophy to the table that you need to workaround.

Angular is an opinionated framework which means that it has its own rules that developers need to learn and follow. This can be a good and a bad thing, but the learning curve can be uncomfortable for total beginners. I would recommend that you stay patient and start slow. Slow and steady!

- Limited SEO Capabilities:

Angular is just great for building powerful single-page web applications. However, as with all single-page web applications, there is a disadvantage when it comes to search engine optimization.

Usually, single-page applications are rendered on the client-side and therefore web crawlers of search engines like Google and Bing are not able to see the complete structure and content of the individual pages of the websites. With this limitation in place,it is not possible to list your website correctly in search engine results.

However, there are techniques and packages to make this work, but it is an extra effort in terms of development. Angular universal is a way to render the Angular application of the server itself. The technology enables server-side rendering for Angular applications. Angular Universal generates static application pages on the server through a process called server-side rendering.

While Angular universal takes care of most aspects, it still requires a lot of setups. If you prefer to do it all manually, there are documents available to help you do that but there are gotchas that you need to watch out for.

- Too many versions:

Angular has been evolving. It started out as a JavaScript framework AngularJS and now the current version is Angular 10. This can cause confusion, especially for beginners.

Many a time, beginners struggle to understand the versioning. I have seen students learning a version of the framework, latest at the time, and by the time they finish their first project, a new major release is out. This causes trust issues. Developers, at times, feel that they must keep working on their apps and resolve version conflicts and compatibility issues.

2.2.6. Angular used in project

In this project, we use Angular 9.1.12 for the entire front-end portion of the project.

2.3. Heroku

Heroku is a fully managed container-based cloud platform, with integrated data services and a powerful ecosystem, for deploying and running modern apps. The Heroku developer experience is based on an app-centric approach to software delivery, and integrates with today's most popular developer tools and workflows.

Heroku runs your apps inside dynos — smart containers on a reliable, fully managed runtime environment. Developers deploy their code written in Node, Ruby, Java, PHP, Python, Go, Scala, or Clojure to a build system which produces an app that's ready for execution. The system and language stacks are monitored, patched, and upgraded, so it's always ready and up-to-date. The runtime keeps apps running without any manual intervention.

Heroku runs your app in lightweight, isolated Linux containers called "dynos." The platform offers different dyno types to help you get the best results for your type of app. With free version, heroku provided any services includes:

- 550-1,000 dyno hours per month.
- Deploy with Git and Docker.
- Custom domains.
- Container orchestration.
- Automatic OS patching.

In the project, users API, promotions API, system API, products API, cdn API, warehouse API are deployed in heroku container.

2.4. Other technologies and libraries

2.4.1. RESTful APIs [8]

REST is acronym for REpresentational State Transfer. It is architectural style for distributed hypermedia systems and was first presented by Roy Fielding in 2000 in his famous dissertation. A RESTful API is an application program interface (API) that uses HTTP requests to GET, PUT, POST and DELETE data.

An API for a website is code that allows two software programs to communicate with each other. The API spells out the proper way for a developer to write a program requesting services from an operating system or other application.

A RESTful API uses existing HTTP methodologies defined by the RFC 2616 protocol. They use GET to retrieve a resource; PUT to change the state of or update a resource, which can be an object, file or block; POST to create that resource; and DELETE to remove it.

Because the calls are stateless, REST is useful in cloud applications. Stateless components can be freely redeployed if something fails, and they can scale to accommodate load changes. This is because any request can be directed to any instance of a component; there can be nothing saved that has to be remembered by the next transaction. That makes REST preferable for web use, but the RESTful model is also helpful in cloud services because binding to a service through an API is a matter of controlling how the URL is decoded. Cloud computing and microservices are almost certain to make RESTful API design the rule in the future.

In the project, all API services are developed base on REST API rules. It's easy to understand, develop and scalable, easy to share the open API to the third third party partners.

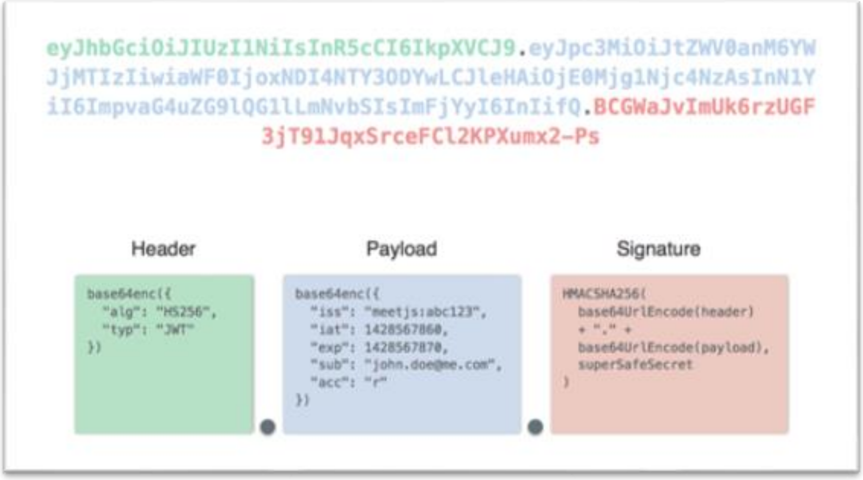
2.4.2. Json web token

JWT (Json Web Token) is an open standard token used to exchange information with HTTP requests. This information is verified and reliably marked based on the signature. JWT has many advantages over sessions.

-Stateless, information is not stored on the server.

-Easy to develop, expand.

-Better performance because the server reads the information in the request (if use session, the system must query information in storage or database).



```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJtZWV0anM6YW
JjMTIzIiwiaWF0IjoxNDI4NTY3ODYwLCJleHAiOjE0Mjg1Njc4NzAsInN1Y
iI6ImpvaG4uZG9lQG1lLmNvbSIsImFjYyI6InIifQ.BCGWaJvImUk6rzUGF
3jT91JqxSrceFCl2KPXumx2-Ps
```

```
base64enc({
  "alg": "HS256",
  "typ": "JWT"
})
```

```
base64enc({
  "iss": "meetjs:abc123",
  "iat": 1428567860,
  "exp": 1428567870,
  "sub": "john.doe@me.com",
  "acc": "r"
})
```

```
HMACSHA256(
  base64UrlEncode(header)
  + "." +
  base64UrlEncode(payload),
  superSafeSecret
)
```

Figure 2.1. Json web token structure

The signature part will be encrypted using HMAC or RSA.

HMAC: The JWT creator (token issuer) and the JWT (token verifier) receiver used the same secret key to encrypt and check.

RSA using 1 key pair, JWT initialization object uses private key to encrypt, JWT receiver uses public key to check.

So that, with HMAC both sides must share a secret key with each other, and the receiver JWT can completely generate another valid JWT code based on that secret key. As for RSA, the receiver uses the public key to check but cannot initiate a new JWT based on that key. Therefore, encryption using RSA makes the signature security better when sharing JWT with many different objects.

In the project, JWT is applied to authentication and authorization.

CHAPTER 3. SURVEY OF STATUS AND DETERMINATION OF REQUIREMENTS

3.1. Survey of current status

Enter the 21st century, with this industry 4.0. English is becoming more and more an indispensable thing for everyone must equip themselves.

English - Has become an essential skill for senior positions in the eyes of home recruitment. Especially in recent years, along with the explosion of big data around the world that is tremendous growth in Artificial Intelligence. That is the reason that each of us must equip ourselves with the skills English if not out of the game is increasingly fierce in this flat world.

For that reason, with the help of the Information Technology 4.0 era, everyone at home has the internet and easy access to the Web with high-speed bandwidth. We need to give English to everyone. At the same time, it creates a favorable environment and facilitates and encourages the self-learning English spirit of each person.

Since then, it helps Vietnamese society to become more and more civilized, and the country of Vietnam is increasingly developing. From there get yourself a position head firmly in this flat world. That is why I chose this topic as my university graduation project.

3.2. Determination of requirements

3.2.1. Functional requirements

3.2.1.1. Professional function requirements

a. Guest:

- **Sign In.**
- **Sign Up.**
- **See Vocabulary Page.**
- **Forgot Password.**

b. User:

- **View Profile Page:**
    - View Account Information.
    - Update Account Information.

    - Change Account Password.
- **View Vocabulary Page.**
- **View Grammar Page.**
- **View Practice Page.**
- **Take Reading Test.**

c. Admin:

- **Manage Vocabularies:**
    - Add Vocabulary.
    - Update Vocabulary.
    - View Vocabulary.
    - Delete Vocabulary.
- **Manage Grammars:**
    - Add Grammar.
    - Update Grammar.
    - View Grammar.
    - Delete Grammar.
- **Manage Practices:**
    - Add Practice.
    - Update Practice.
    - View Practice.
    - Delete Practice.
- **Manage Users:**
    - Add User.
    - Update User.
    - View User.
    - Delete User.
- **Manage QuestionReadings:**
    - Add QuestionReading.
    - Update QuestionReading.
    - View QuestionReading.
    - Delete QuestionReading.
- **Manage Paragraphs:**
    - Add Paragraph.
    - Update Paragraph.
    - View Paragraph.
    - Delete Paragraph.

3.2.1.2. System function requirements

- **Authorization:**
    - Admin: Have all the role of the system.
    - User: Take the role of a member.
    - Guest: No role. Anonymous.
- **Notification:**
    - Always be informed when the operation succeeds or fails.
- **Security:**
    - Always check the permissions before working with data.
    - Against basic attacks such as: XSS, Sql Injection, …
- **Storage:**
    - Images
    - Vocabularys
    - Grammars
    - Practices
    - QuestionReadings
    - Paragraphs
    - Users

3.2.2. Non-functional requirements

- **Usability:**
    - Easy-to-see, user-friendly interface.
    - Easy to manipulate.
- **Effectiveness:**
    - Fast and efficient search speeds.
    - Fast page loading speed.
- **Compatibility:**
    - To meet multiple platforms and devices.
- **Development:**
    - Easy to develop applications.

CHAPTER 4. REQUIREMENT MODELING

4.1. Define usecase

4.1.1. Actor

Table 4.1. Actor

No	Use Case	Admin Actor	User Actor	Guest Actor
01	Sign Up			x
02	Sign In			x
03	Forgot Password			x
04	Sign Out	x	x	x
05	View Profile Page	x	x	
06	View Vocabulary Page	x	x	
07	View Grammar Page	x	x	
08	View Practice Page	x	x	
09	Take Reading Test	x	x	
10	Manage Vocabularies	x		
11	Manage Grammars	x		
12	Manage Practices	x		
13	Manage Users	x		
14	Manage QuestionReadings	x		
15	Manage Paragraphs	x		

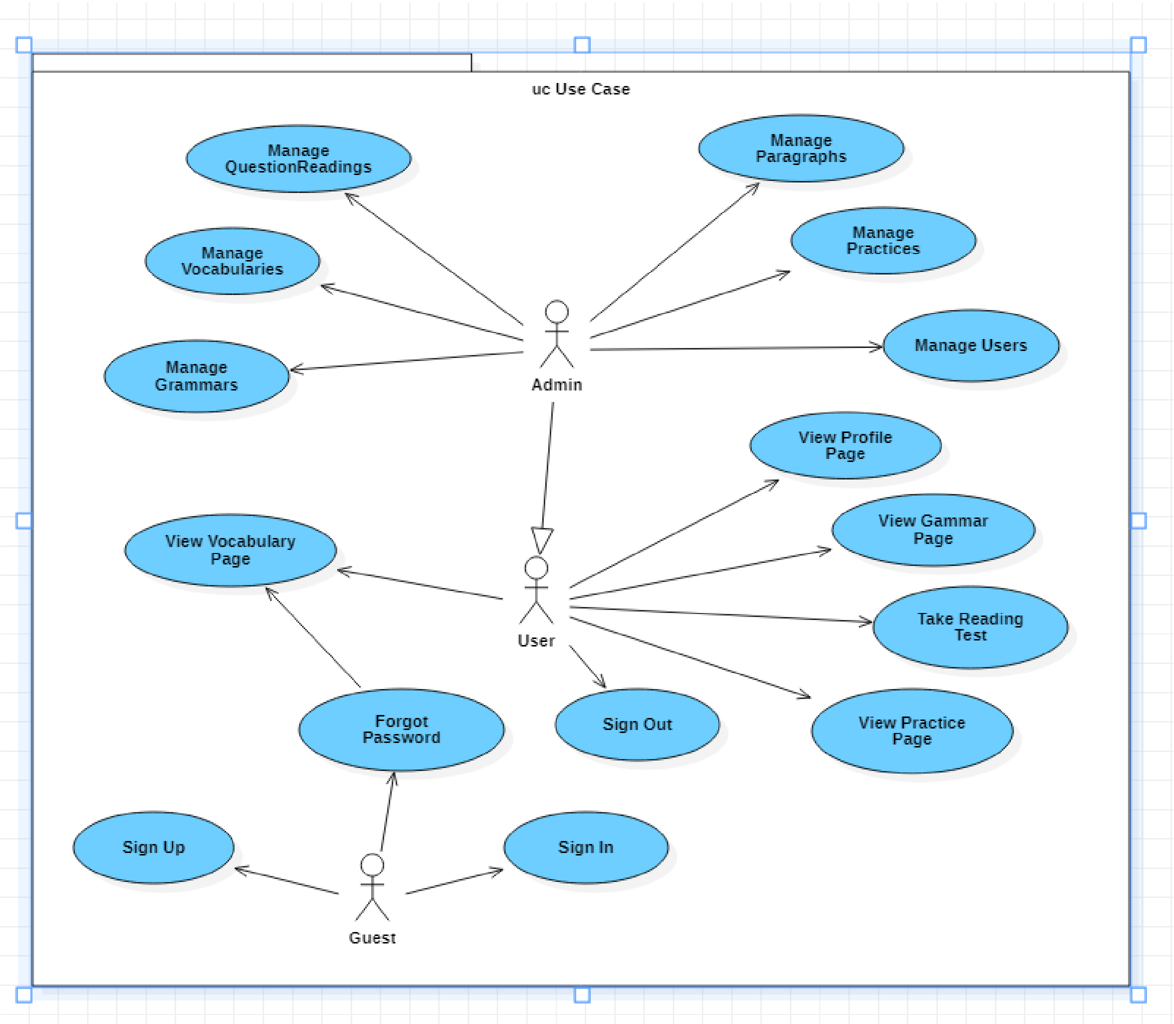
4.1.2. Use case

Table 4.2. Use Case

No	Use Case Name		UseCase_ID
01	Sign Up		UC_1
02	Sign In		UC_2
	2.1	Sign In With Google	UC_2.1
	2.2	Sign In With Account	UC_2.2
03	Forgot Password		UC_3
04	Sign Out		UC_4
05	View Profile Page		UC_5
	5.1	View Account Information	UC_5.1
	5.2	Update Account Information	UC_5.2
	5.3	Change Account Password	UC_5.3
06	View Vocabulary Page		UC_6
07	View Grammar Page		UC_7
08	View Practice Page		UC_8
09	Take Reading Test		UC_9
10	Manage Vocabularies		UC_10
	10.1	Add Vocabulary	UC_10.1
	10.2	Update Vocabulary	UC_10.2
	10.3	View Vocabularies	UC_10.3
	10.4	Delete Vocabulary	UC_10.4
11	Manage Grammars		UC_11
	11.1	Add Grammar	UC_11.1
	11.2	Update Grammar	UC_11.2
	11.3	View Grammars	UC_11.3

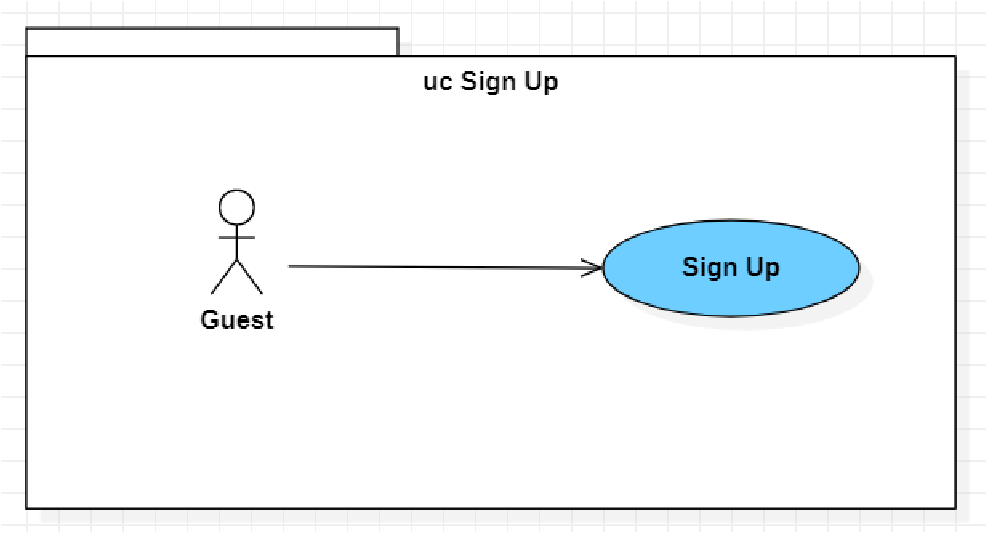
	11.4	Delete Grammar	UC_11.4
12	Manage Practices		UC_12
	12.1	Add Practice	UC_12.1
	12.2	Update Practice	UC_12.2
	12.3	View Practices	UC_12.3
	12.4	DeletePractice	UC_12.4
13	Manage Users		UC_13
	13.1	Add User	UC_13.1
	13.2	Update User	UC_13.2
	13.3	View User	UC_13.3
	13.4	Delete User	UC_13.4
14	Manage QuestionReadings		UC_14
	14.1	Add QuestionReadings	UC_14.1
	14.2	Update QuestionReadings	UC_14.2
	14.3	View QuestionReadings	UC_14.3
	14.4	Delete QuestionReadings	UC_14.4
15	Manage Paragraphs		UC_15
	15.1	Add Paragraphs	UC_15.1
	15.2	Update Paragraphs	UC_15.2
	15.3	View Paragraphs	UC_15.3
	15.4	Delete Paragraphs	UC_15.4

4.2. Use case diagram


uc Use Case
Manage QuestionReadings
Manage Paragraphs
Manage Practices
Manage Vocabularies
Manage Grammars
Manage Users
Admin
View Profile Page
View Vocabulary Page
View Gammar Page
User
Take Reading Test
Forgot Password
Sign Out
View Practice Page
Sign Up
Sign In
Guest

4.3. Use case specification

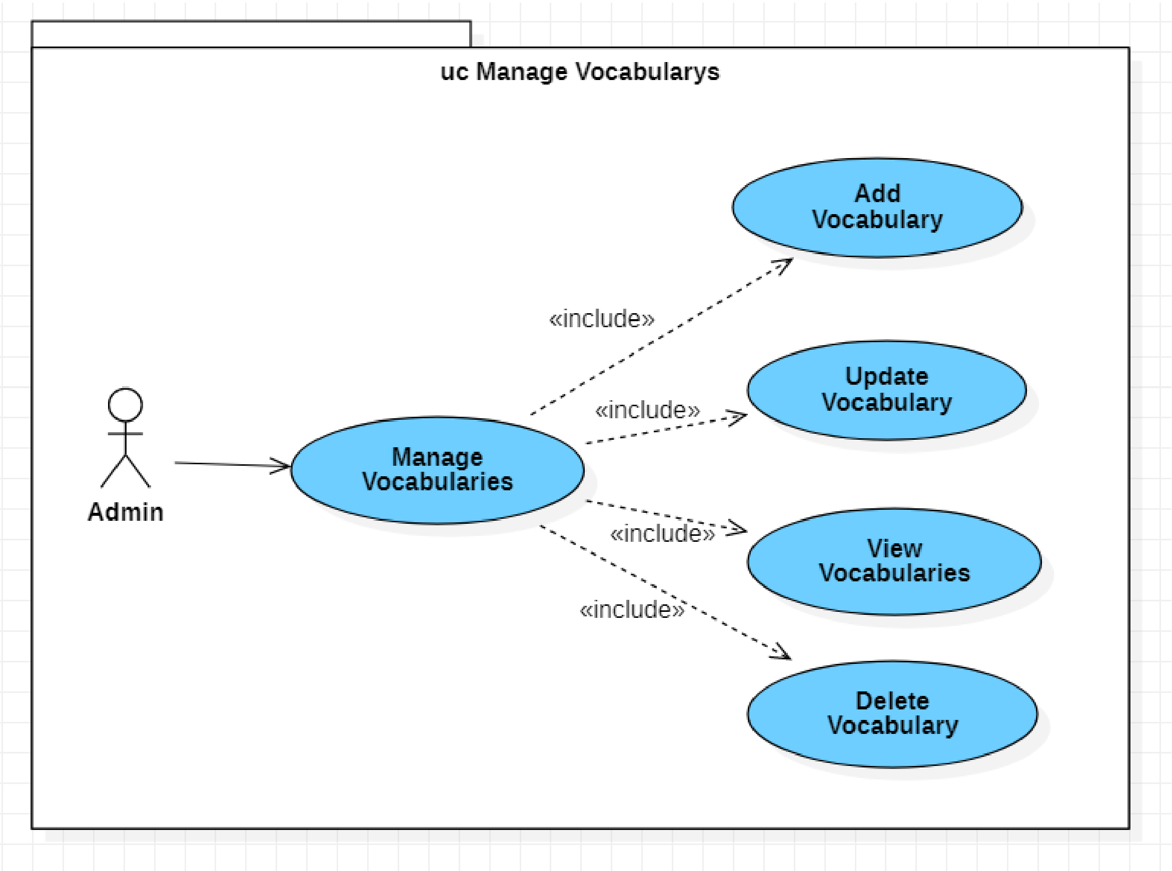
4.3.1. Sign Up:



Use Case ID	UC_1
Name	Sign Up
Goal	Register an account on the system as a member.
Actors	Guest
Pre-conditions	Internet connection available.
Main flow	**(1)** In the interface Home page, Actors click on the icon Sign Up to go to the Sign Up Page. **(2)** In the Registration page, enter the valid values fields: **(3)** Click the Sign Up button. **(4)** If all values are valid: - If all values are valid, the system will give two methods of account authentication: + Authenticate by Email. + Authenticate with OTP code from phone. - The system sends a confirmation code to the user according to the user chosen method. - The system requires a verification code from the user. - The user receives a verification code from the system and enters it.

	- If the verification code is correct, the system will proceed to create a new account. - If the verification code is correct, the system will proceed to create a new account. Give a message "Account creation successful". Link to Home Page. - If the verification code is wrong, the system will request to re-enter the verification code. **(5)** If the value of a certain field is not valid: - The system keeps the user input form and announces the cause of the error below.
Exception	N/a.
Open Issues	N/a.

4.3.2. Manage Vocabularies:



4.3.2.1. Add Vocabulary:

Use Case ID	UC_15.1
Name	Add Vocabulary
Goal	Add a vocabulary to the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.
Main flow	1. In the Home interface, Actor click on the Admin icon, then select Vocabulary to go to the Vocabulary page. 2. On the Vocabulary page, Actor click the Create button. Then will be forwarded to the Create Vocabulary page. 3. On the Create Vocabulary page, Actor filled out the information. Then click on the Save button: - If all are valid, the new Vocabulary will be successfully created. The system generate the message "Created successfully". The web page will be forwarded to the Vocabulary page. - If something is not valid, creating new Vocabulary will fail. The system will keep the form and give errors. 4. In the process of filling in information in the new Vocabulary form, Actor can cancel at any time by clicking on the Cancel button. The system will return to the previous state in the Vocabulary page. 5. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.2.2. *Update Vocabulary:*

Use Case ID	UC_10.2
Name	Update Vocabulary
Goal	Update a vocabulary to the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin

Main flow	1. In the Home interface, Actor click on the Admin icon, then select Vocabulary to go to the Vocabulary page. 2. On the Vocabulary page, Actor select the vocabulary to update. Then click on the Update button to update the selected vocabulary. The web page will forward to the UpdateVocabulary page. 3. On the UpdateVocabulary page, it will display the form with the current information of the selected vocabulary in the database. Actor updates vocabulary information. Then click the Save button: - If all is valid, updating the vocabulary information to the database will be successful. The system will generate the message "Updated Successful". The web page will forward to the Paragraph page. - If something is not valid, updating the vocabulary information to the database will fail. The system will keep the form and give errors. 4. During the process of filling in the form vocabulary update information, Actor can cancel at any time by clicking on the Cancel button. The system will return to the previous state in the Vocabulary page. 5. Usecase ends the session.
Exception	N/a
Open Issues	N/a

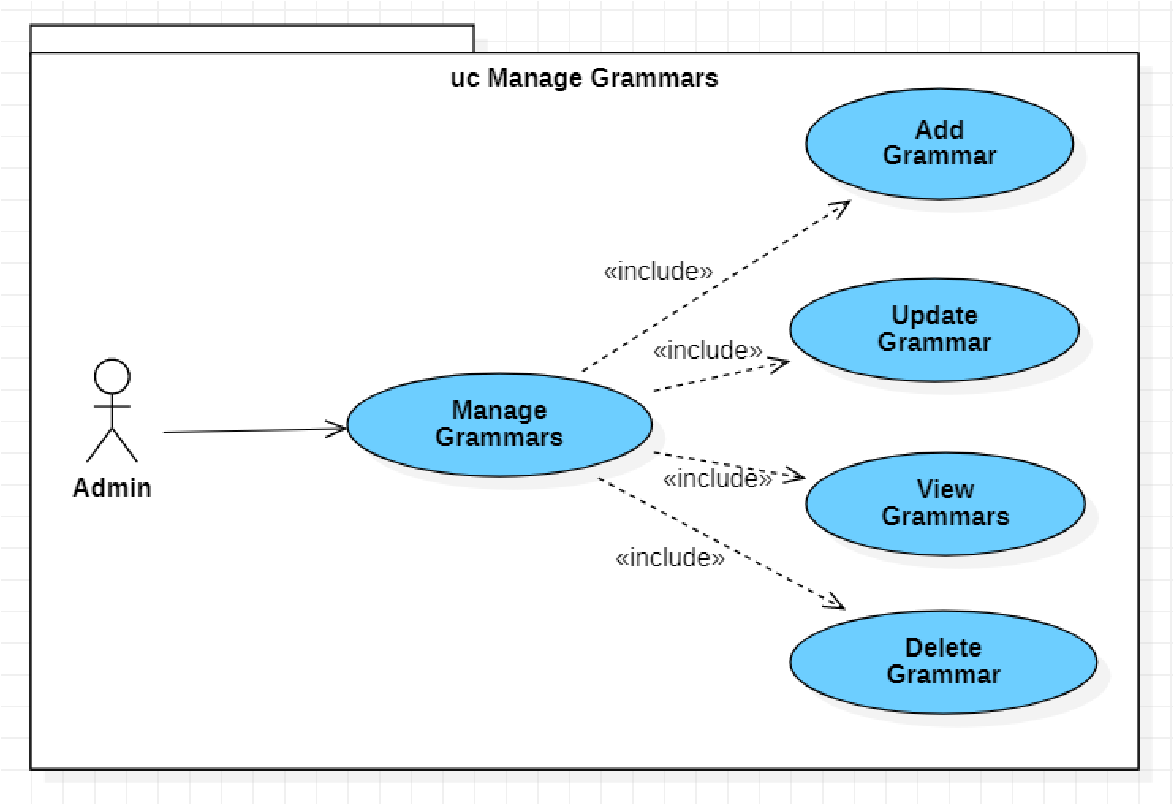
4.3.2.3. *View Vocabularies:*

Use Case ID	UC_10.3
Name	View Vocabularies
Goal	View all vocabularies currently available in Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin
Main flow	1. In the Home interface, Actor click on the Admin icon, then select Vocabulary to go to the Vocabulary page. 2. Actors will view the Vocabularies here. 3. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.2.4. *Delete Vocabulary:*

Use Case ID	UC_15.4
Name	Delete Vocabulary
Goal	Delete a vocabulary from the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.
Main flow	1. In the Home interface, Actor click on the Admin icon, then select Vocabulary to go to the Vocabulary page. 2. In the Vocabulary page, Actor select the Vocabulary to delete. Then click the Delete button to delete the selected Vocabulary. 3. Does the system give a confirmation message that Actor wants to delete? - If Actor accepts, the Vocabulary will be removed and will reload the Vocabulary Page. - If the Actor clicks Cancel, the Vocabulary Action will be canceled and the Actor will return to its previous state. 4. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.3. Manage Grammars:



4.3.3.1. Add Grammar:

Use Case ID	UC_15.1
Name	Add Grammar
Goal	Add a grammar to the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.

Main flow	**(1)** In the Home interface, Actor click on the Admin icon, then select Grammar to go to the Grammar page. **(2)** On the Grammar page, Actor click the Create button. Then will be forwarded to the Create Grammar page. **(3)** On the Create Grammar page, Actor filled out the information. Then click on the Save button: - If all are valid, the new Grammar will be successfully created. The system generate the message "Created successfully". The web page will be forwarded to the Grammar page. - If something is not valid, creating new Grammar will fail. The system will keep the form and give errors. **(4)** In the process of filling in information in the new Grammar form, Actor can cancel at any time by clicking on the Cancel button. The system will return to the previous state in the Grammar page. **(5)** Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.3.2. Update Grammar:

Use Case ID	UC_11.2
Name	Update Grammar
Goal	Update a grammar to the Database
Actors	Admin
Pre-conditions	Has logged into the system as an admin
Main flow	1. In the Home interface, Actor click on the Admin icon, then select Grammar to go to the Grammar page. 2. On the Grammar page, Actor select the grammar to update. Then click on the Update button to update the selected grammar. The web page will forward to the UpdateGrammar page. 3. On the UpdateGrammar page, it will display the form with the current information of the selected grammar in the database.

	Actor updates grammar information. Then click the Save button: - If all is valid, updating the grammar information to the database will be successful. The system will generate the message "Updated Successful". The web page will forward to the Grammar page. - If something is not valid, updating the grammar information to the database will fail. The system will keep the form and give errors. 4. During the process of filling in the form grammar update information, Actor can cancel at any time by clicking on the Cancel button. The system will return to the previous state in the Grammar page. 5. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.3.3. *View Grammars:*

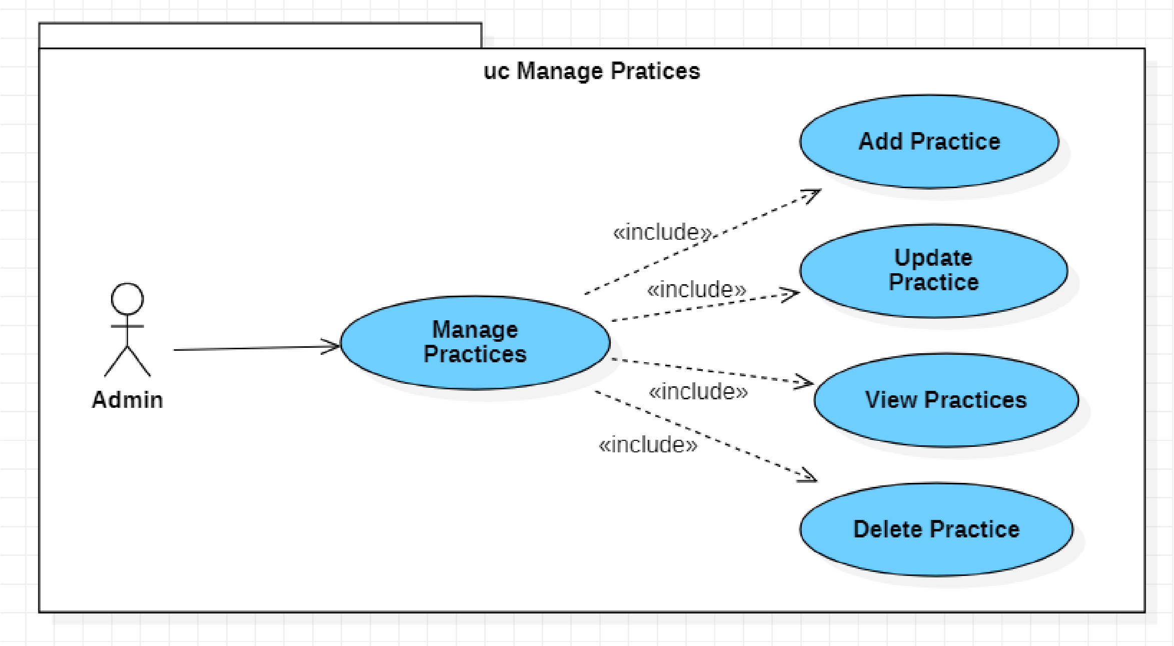
Use Case ID	UC_11.3
Name	View Grammars
Goal	View all grammars currently available in Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin
Main flow	1. In the Home interface, Actor click on the Admin icon, then select Grammar to go to the Grammar page. 2. Actors will view the Grammars here. 3. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.3.4. *Delete Grammar:*

Use Case ID	UC_15.4
Name	Delete Grammar
Goal	Delete a grammar from the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.

Main flow	1. In the Home interface, Actor click on the Admin icon, then select Grammar to go to the Grammar page. 2. In the Grammar page, Actor select the Grammar to delete. Then click the Delete button to delete the selected Grammar. 3. Does the system give a confirmation message that Actor wants to delete? - If Actor accepts, the Grammar will be removed and will reload the Grammar Page. - If the Actor clicks Cancel, the Grammar Action will be canceled and the Actor will return to its previous state. 4. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.4. Manage Practices:



4.3.4.1. Add Practice:

Use Case ID	UC_15.1
Name	Add Practice
Goal	Add a practice to the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.

Main flow	1. In the Home interface, Actor click on the Admin icon, then select Practice to go to the Practice page. 2. On the Practice page, Actor click the Create button. Then will be forwarded to the Create Practice page. 3. On the Create Practice page, Actor filled out the information. Then click on the Save button: - If all are valid, the new Practice will be successfully created. The system generate the message "Created successfully". The web page will be forwarded to the Practice page. - If something is not valid, creating new Practice will fail. The system will keep the form and give errors. 4. In the process of filling in information in the new Practice form, Actor can cancel at any time by clicking on the Cancel button. The system will return to the previous state in the Practice page. 5. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.4.2. Update Practice:

Use Case ID	UC_12.2
Name	Update Practice
Goal	Update a practice to the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin
Main flow	1. In the Home interface, Actor click on the Admin icon, then select Practice to go to the Practice page. 2. On the Practice page, Actor select the practice to update. Then click on the Update button to update the selected practice. The web page will forward to the UpdatePractice page. 3. On the UpdatePractice page, it will display the form with the current information of the selected practice in the database. Actor updates practice information. Then click the Save button: - If all is valid, updating the practice information to the database will be successful. The system will generate the message "Updated Successful". The web page will forward to the Practice page.

	- If something is not valid, updating the practice information to the database will fail. The system will keep the form and give errors. 4. During the process of filling in the form practice update information, Actor can cancel at any time by clicking on the Cancel button. The system will return to the previous state in the Practice page. 5. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.4.3. View Practices:

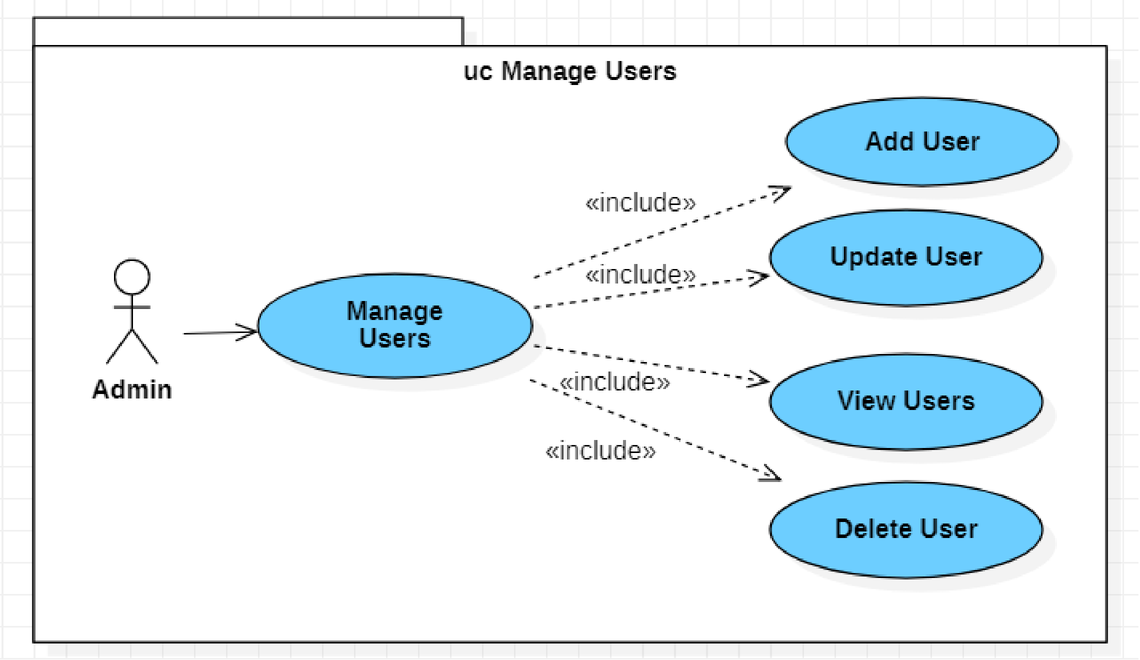
Use Case ID	UC_12.3
Name	View Practices
Goal	View all practices currently available in Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin
Main flow	1. In the Home interface, Actor click on the Admin icon, then select Practice to go to the Practice page. 2. Actors will view the Practices here. 3. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.4.4. Delete Practice:

Use Case ID	UC_15.4
Name	Delete Practice
Goal	Delete a practice from the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.

Main flow	1. In the Home interface, Actor click on the Admin icon, then select Practice to go to the Practice page. 2. In the Practice page, Actor select the Practice to delete. Then click the Delete button to delete the selected Practice. 3. Does the system give a confirmation message that Actor wants to delete? - If Actor accepts, the Practice will be removed and will reload the Practice Page. - If the Actor clicks Cancel, the Practice Action will be canceled and the Actor will return to its previous state. 4. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.5. Manage Users:



4.3.5.1. Add User:

Use Case ID	UC_15.1
Name	Add User
Goal	Add a user to the Database
Actors	Admin

Pre-conditions	Has logged into the system as an Admin.
Main flow	1. In the Home interface, Actor click on the Admin icon, then select User to go to the User page. 2. On the User page, Actor click the Create button. Then will be forwarded to the Create User page. 3. On the Create User page, Actor filled out the information. Then click on the Save button: - If all are valid, the new User will be successfully created. The system generate the message "Created successfully". The web page will be forwarded to the User page. - If something is not valid, creating new User will fail. The system will keep the form and give errors. 4. In the process of filling in information in the new User form, Actor can cancel at any time by clicking on the Cancel button. The system will return to the previous state in the User page. 5. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.5.2. Update User:

Use Case ID	UC_13.2
Name	Update User
Goal	Update a user to the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin
Main flow	1. In the Home interface, Actor click on the Admin icon, then select User to go to the User page. 2. On the User page, Actor select the user to update. Then click on the Update button to update the selected user. The web page will forward to the Update User page. 3. On the Update User page, it will display the form with the current information of the selected user in the database. Actor updates user information. Then click the Save button: - If all is valid, updating the paragraph information to the database will be successful. The system will generate

	the message "Updated Successful". The web page will forward to the User page. - If something is not valid, updating the user information to the database will fail. The system will keep the form and give errors. 4. During the process of filling in the form user update information, Actor can cancel at any time by clicking on the Cancel button. The system will return to the previous state in the User page. 5. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.5.3. *View Users:*

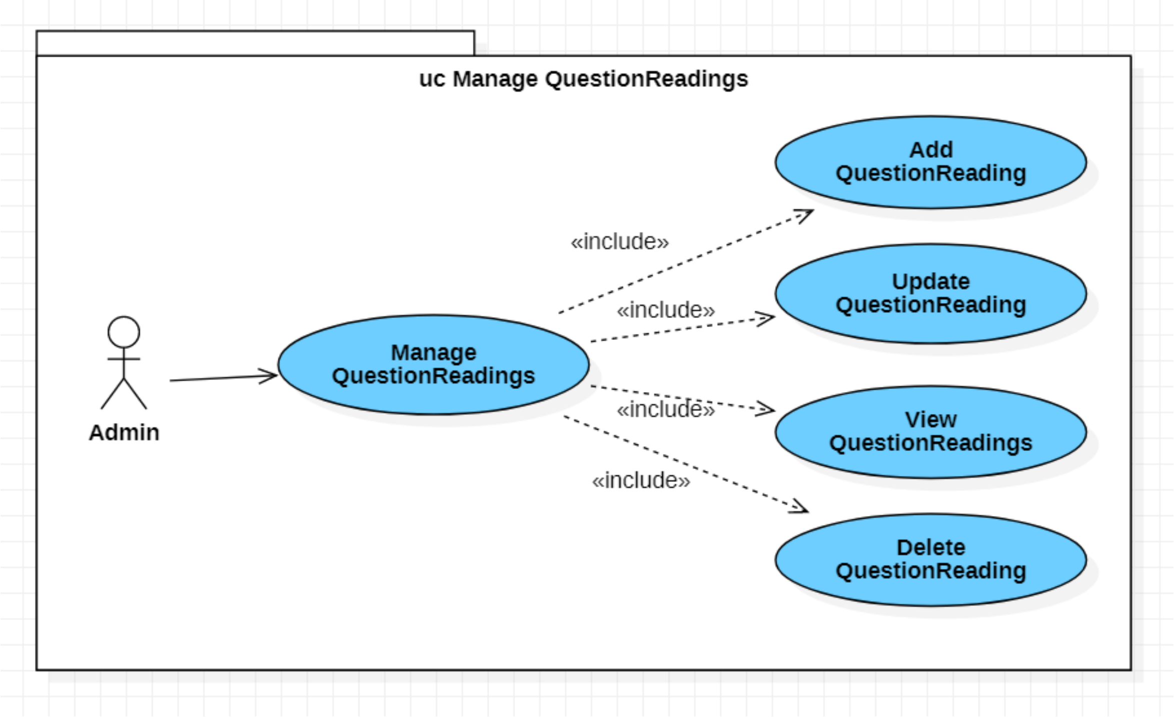
Use Case ID	UC_13.3
Name	View Users
Goal	View all users currently available in Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin
Main flow	1. In the Home interface, Actor click on the Admin icon, then select User to go to the User page. 2. Actors will view the User here. 3. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.5.4. *Delete User:*

Use Case ID	UC_15.4
Name	Delete User
Goal	Delete a user from the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.

Main flow	1. In the Home interface, Actor click on the Admin icon, then select User to go to the User page. 2. In the User page, Actor select the User to delete. Then click the Delete button to delete the selected User. 3. Does the system give a confirmation message that Actor wants to delete? - If Actor accepts, the User will be removed and will reload the User Page. - If the Actor clicks Cancel, the User Action will be canceled and the Actor will return to its previous state. 4. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.6. Manage QuestionReadings:



4.3.6.1. Add QuestionReading:

Use Case ID	UC_15.1
Name	Add QuestionReading
Goal	Add a question reading to the Database

Actors	Admin
Pre-conditions	Has logged into the system as an Admin.
Main flow	**1.** In the Home interface, Actor click on the Admin icon, then select QuestionReading to go to the QuestionReading page. **2.** On the QuestionReading page, Actor click the Create button. Then will be forwarded to the Create QuestionReading page. **3.** On the Create QuestionReading page, Actor filled out the information. Then click on the Save button: - If all are valid, the new QuestionReading will be successfully created. The system generate the message "Created successfully". The web page will be forwarded to the QuestionReading page. - If something is not valid, creating new QuestionReading will fail. The system will keep the form and give errors. **4.** In the process of filling in information in the new QuestionReading form, Actor can cancel at any time by clicking on the Cancel button. The system will return to the previous state in the QuestionReading page. **5.** Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.6.2. Update QuestionReading:

Use Case ID	UC_15.2
Name	Update QuestionReading
Goal	Update a question reading to the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.

Main flow	**1.** In the Home interface, Actor click on the Admin icon, then select QuestionReading to go to the QuestionReading page. **2.** On the QuestionReading page, Actor select the question reading to update. Then click on the Update button to update the selected question reading. The web page will forward to the UpdateQuestionReading page. **3.** On the UpdateQuestionReading page, it will display the form with the current information of the selected question reading in the database. Actor updates question reading information. Then click the Save button: - If all is valid, updating the question reading information to the database will be successful. The system will generate the message "Updated Successful". The web page will forward to the QuestionReading page. - If something is not valid, updating the question reading information to the database will fail. The system will keep the form and give errors. **4.** During the process of filling in the form question reading update information, Actor can cancel at any time by clicking on the Cancel button. The system will return to the previous state in the QuestionReading page. **5.** Usecase ends the session.
Exception	N/a
Open Issues	N/a

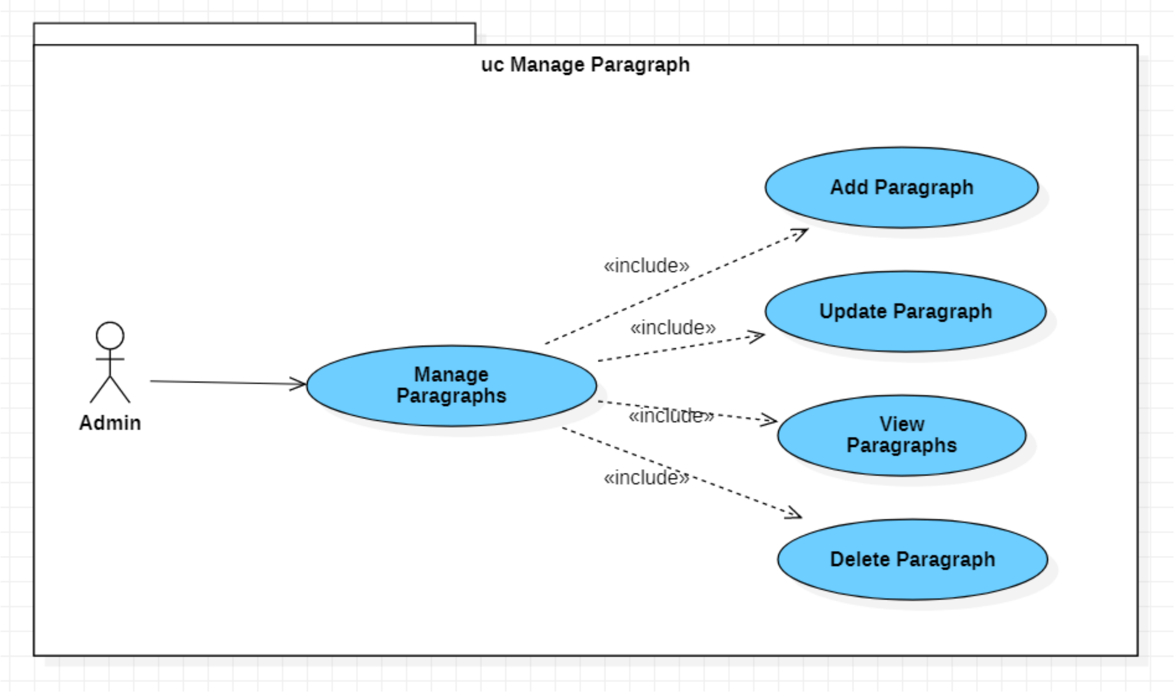
4.3.6.3. View QuestionReadings:

Use Case ID	UC_14.3
Name	View QuestionReadings
Goal	View all toeic tests currently available in Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin
Main flow	1. In the Home interface, Actor click on the Admin icon, then select QuestionReading to go to the QuestionReading page. 2. Actors will view the ToeicTests here. 3. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.6.4. Delete QuestionReadings:

Use Case ID	UC_15.4
Name	Delete QuestionReading
Goal	Delete a question reading from the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.
Main flow	1. In the Home interface, Actor click on the Admin icon, then select QuestionReading to go to the QuestionReading page. 2. In the QuestionReading page, Actor select the QuestionReading to delete. Then click the Delete button to delete the selected QuestionReading. 3. Does the system give a confirmation message that Actor wants to delete? - If Actor accepts, the QuestionReading will be removed and will reload the QuestionReading Page. - If the Actor clicks Cancel, the QuestionReading Action will be canceled and the Actor will return to its previous state. 4. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.7. Manage Paragraphs



4.3.7.1. Add Paragraph

Use Case ID	UC_15.1
Name	Add Paragraph
Goal	Add a paragraph to the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.
Main flow	**1.** In the Home interface, Actor click on the Admin icon, then select Paragraph to go to the Paragraph page. **2.** On the Paragraph page, Actor click the Create button. Then will be forwarded to the Create Paragraph page. **3.** On the Create Paragraph page, Actor filled out the information. Then click on the Save button: - If all are valid, the new Paragraph will be successfully created. The system generate the message "Created successfully". The web page will be forwarded to the Paragraph page. - If something is not valid, creating new Paragraph will fail. The system will keep the form and give errors. **4.** In the process of filling in information in the new Paragraph form, Actor can cancel at any time by clicking on the Cancel button. The system will return to the previous state in the Paragraph page. **5.** Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.7.2. Update Paragraph

Use Case ID	UC_15.2
Name	Update Paragraph
Goal	Update a paragraph to the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.
Main flow	1. In the Home interface, Actor click on the Admin icon, then select Paragraph to go to the Paragraph page. 2. On the Paragraph page, Actor select the paragraph to update. Then click on the Update button to update the selected paragraph. The web page will forward to the UpdateParagraph page. 3. On the UpdateParagraph page, it will display the form with the current information of the selected paragraph in the

	database. Actor updates paragraph information. Then click the Save button: - If all is valid, updating the paragraph information to the database will be successful. The system will generate the message "Updated Successful". The web page will forward to the Paragraph page. - If something is not valid, updating the paragraph information to the database will fail. The system will keep the form and give errors. 4. During the process of filling in the form paragraph update information, Actor can cancel at any time by clicking on the Cancel button. The system will return to the previous state in the Paragraph page. 5. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.7.3. *View Paragraphs*

Use Case ID	UC_15.3
Name	View Paragraphs
Goal	View paragraphs from the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.
Main flow	1. In the Home interface, Actor click on the Admin icon, then select Paragraph to go to the Paragraph page. 2. Actors will view the Paragraphs here. 3. Usecase ends the session.
Exception	N/a
Open Issues	N/a

4.3.7.4. *Delete Paragraph*

Use Case ID	UC_15.4
Name	Delete Paragraph
Goal	Delete a paragraph from the Database
Actors	Admin
Pre-conditions	Has logged into the system as an Admin.

Main flow	1. In the Home interface, Actor click on the Admin icon, then select Paragraph to go to the Paragraph page. 2. In the Paragraph page, Actor select the Paragraph to delete. Then click the Delete button to delete the selected Paragraph. 3. Does the system give a confirmation message that Actor wants to delete? - If Actor accepts, the Paragraph will be removed and will reload the Paragraph Page. - If the Actor clicks Cancel, the Paragraph Action will be canceled and the Actor will return to its previous state. 4. Usecase ends the session.
Exception	N/a
Open Issues	N/a

CHAPTER 5. SOFTWARE DESIGN

5.1. Database Design

5.1.1. AspNetRoles

Purpose: Save the roles in the system.

No	Column	Type	Key	Allow Null	Description
1	Id	INTEGER	PK		Save the Id of each role object, as the primary key, must not be null and increment automatically.
2	Name	TEXT			The name of the role powers in the application.
3	NormalizedName	TEXT			
4	ConcurrencyStamp	TEXT			

5.1.2. AspNetUserRoles

Purpose: A table created from a many-to-many association between the users table and the roles table. Store UserId and RoleId.

No	Column	Type	Key	Allow Null	Description
1	UserId	INTEGER	PK		Is one of the primary keys of the table. Store User's Id. Must not be null.
2	RoleId	INTEGER	PK		Is one of the primary keys of the table. Store the Role Id. Must not be null.

5.1.3. AspNetUsers

Purpose: Store account information and personal information of users in the system.

No	Column	Type	Key	Allow Null	Description
1	Id	INTEGER	PK		Save the Id of each user object, as the primary key, must not be null and increment automatically.
2	Username	TEXT			Username used in the application.
3	NormalizedUserName	TEXT			
4	Email	TEXT		✓	Email registration of the user's account.
5	PasswordHash	TEXT			The user's password is encrypted.
6	SecurityStamp	TEXT			
7	ConcurrencyStamp	TEXT			
8	DateOfBirth	TEXT		✓	User's date of birth.
9	KnownAs	TEXT			User aliases used in the application.
10	PhotoUrl	TEXT		✓	Url leads to the user's profile picture.
11	Gender	TEXT		✓	User's gender.
12	City	TEXT		✓	The city where the user lives.
13	Country	TEXT		✓	The country in which the user lives.

5.1.4. Vocabulary

Purpose: Save Vocabulary objects

No	Column	Type	Key	Allow Null	Description
1	Id	INTEGER	PK		Save the Id of each Vocabulary object, as the primary key, must not be null and increment automatically.
2	VietName	TEXT			Vietnamese meaning of vocabulary.
3	EngName	TEXT			The English meaning of the vocabulary.
4	Image	TEXT			Visual images illustrate vocabulary.
5	Description	TEXT			Definition, description, word type of the vocabulary.

5.1.5. Grammar

Purpose: Save Grammar Objects.

No	Column	Type	Key	Allow Null	Description
1	Id	INTEGER	PK		Save the Id of each Grammar object, as the primary key, must not be null and increment automatically.
2	GrammarName	TEXT			Grammar name.
3	Structure	TEXT			The content of the grammatical structure
4	Description	TEXT			Definition, description, usage of grammatical structure.

5	Example	TEXT			Some good examples apply grammatical structures.

5.1.6. Practice

Purpose: Save Practices Objects

No	Column	Type	Key	Allow Null	Description
1	Id	INTEGER	PK		Save the Id of each Practice object, as the primary key, must not be null and increment automatically.
2	PracticeName	TEXT			The category names of the exercises.
3	Question	TEXT			The content of the question.
4	AnswerA	TEXT			Option A is for the question.
5	AnswerB	TEXT			Option B is for the question.
6	AnswerC	TEXT			Option C is for the question.
7	AnswerD	TEXT			Option D is for the question.
8	Answer	TEXT			Correct answer of the question.

5.1.7. Paragraph

Purpose: Save Paragraphs Objects

No	Column	Type	Key	Allow Null	Description
1	Id	INTEGER	PK		Save the Id of each Paragraph object, as the primary key, must not be null and

					increment automatically.
2	ToeicNumber	TEXT			Toeic Test Number.
3	ToeicPart	TEXT			Part number in Toeic test.
4	QuestionNumber	TEXT			Number of questions in Toeic test.
5	ParagraphText	TEXT			The text is for reading comprehension to answer the corresponding questions on the Toeic test.

5.1.8. QuestionReading

Purpose: Save QuestionReading Objects

No	Column	Type	Key	Allow Null	Description
1	Id	INTEGER	PK		Save the Id of each QuestionReading object, as the primary key, must not be null and increment automatically.
2	ToeicNumber	TEXT			Toeic Test Number.
3	ToeicPart	TEXT			Part number in Toeic test.
4	QuestionNumber	TEXT			Number of questions in Toeic test.
5	Question	TEXT			The content of the question.
6	AnswerA	TEXT			Option A is for the question.
7	AnswerB	TEXT			Option B is for the question.

8	AnswerC	TEXT			Option C is for the question.
9	AnswerD	TEXT			Option D is for the question.
10	Answer	TEXT			Correct answer of the question.

CHAPTER 6. INSTALLATION AND TESTING

6.1. Installation

6.1.1. Production environment

6.1.1.1. Libraries and software need

Libraries and software need to be installed in production:

No	Library/Software	Download Url
1	.NET Core 3.1 SDK	https://dotnet.microsoft.com/download/dotnet-core/thank-you/sdk-3.1.404-windows-x64-installer
2	Node JS 14.15.4 LTS	https://nodejs.org/en/
3	SQLite 3.34.0	https://www.sqlite.org/download.html

6.2. Development environment

6.2.1. Libraries and software need

Libraries and software need to be installed in development:

No	Library/Software	Download Url
1	.NET Core 3.1 SDK	https://dotnet.microsoft.com/download/dotnet-core/thank-you/sdk-3.1.404-windows-x64-installer
2	Node JS 14.15.4 LTS	https://nodejs.org/en/
3	Angular CLI 9.1.13	https://www.npmjs.com/package/@angular/cli/v/9.1.13
4	SQLite 3.34.0	https://www.sqlite.org/download.html

6.2.2. List of command for install and run projects

List of command for install and run projects

No	Command	Description
1	dotnet run	Run ASP.NET Core project (includes API Services and Gateway Services)
2	npm install	Install all libraries of project using for Angular project.
3	ng serve --open	Start project using for Angular project.

6.3. Testing

6.3.1. Admin

6.3.1.1. Test case function CreateVocabulary

Test Case ID	CV_001	**Test Case Description**	Test the functionality to create a new Vocabulary object.		
Created By	Lam	**Reviewed By**		Version	1.1

QA Tester's Log	Review comments from Lam incorprate in version 1.0

Tester's Name	Lam	**Date Tested**	29/12/2020	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Visit Google Chrome
2	Go to Url: http: // localhost: 4200
3	Log in to the website with an admin account with Admin rights.

S #	Test Data
1	VietName = "Xin chào"
2	EngName = "Hello"
3	Image = "https://firebasestorage.googleapis.com/v0/b/englishweb-firebase.appspot.com/o/uploads%2Fhello.jpg?alt=media&token=7755823d-a28f-4002-8a3e-3dbf3e7b9afd"

4	Accessed the URL: http://localhost:4200/admin/vocabulary/createvocabulary	4	Description = "Được dùng để chào hỏi. Thường là câu nói đầu trong giao tiếp".

Test Scenario	Verifying valid input data, admin can create a new Vocabulary object and store it in Database.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Fill in the corresponding values in the Form: VietName, EngName, Image, Description.	Information is entered and displayed correctly.	As Expected	Pass
2	Click button Create.	The message "Created successfully" will appear. Forward to the vocabulary view page.	As Expected	Pass
3				
4				

6.3.2. User

6.3.2.1. Test case function GetVocabularies

Test Case ID	GV_	**Test Case Description**	Test display function of words that can be learned in the system.

	001				
Created By	L	**Reviewed By**	Lam	**Version**	1.1

QA Tester's Log	Review comments from Lam incorprate in version 1.0

Tester's Name	Lam	**Date Tested**	29/12/2020	**Test Case (Pass/Fail/Not Executed)**	Pass

S #	Prerequisites:
1	Visit Google Chrome
2	Go to Url: http: // localhost: 4200
3	Signed in to the site.
4	

S #	Test Data
1	PageNumber = "1"
2	PageSize = 10
3	
4	

Test Scenario	Verify valid input data, display possible vocabulary in the system.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Go to the URL: http://localhost:4200/admin/vocabulary/getvocabularies	Display the number of vocabulary information as 10 if currently in the Database there are at least 10 words, otherwise show	As Expected	Pass

		all the information of the current vocabulary in the Database.		
2				
3				
4				

6.3.3. Guest

6.3.3.1. Test case funtion Register

Test Case ID	R_001	**Test Case Description**	Test account registration functionality		
Created By	Lam	**Reviewed By**	Lam	Version	1.1

QA Tester's Log	Review comments from Lam incorprate in version 1.0

Tester's Name	Lam	**Date Tested**	July 1, 2020	Test Case (Pass/Fail/Not Executed)	Pass

S #	Prerequisites:
1	Visit Google Chrome
2	Go to Url: http: // localhost: 4200
3	Click the Register button on the NavBar
4	

S #	Test Data
1	Username = "tommy"
2	KnownAs = "Tom"
3	Gender = "male"
4	PhotoUrl = ""
5	DateOfBirth = ""
6	City = "Ha Noi"
7	Country = "Viet Nam"
8	Password = "Password1"

Test Scenario	Verify the entered data is valid, the user successfully registered the account.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Fill in the corresponding values in the Form: UserName, KnownAs, Gender, PhotoUrl, DateOfBirth, City, Country, Password.	Information is entered and displayed correctly.	As Expected	Pass
2	Click Button Register	The message "Register for an Account Success!" Appears. Forward to Home page.	As Expected	Pass
3				
4				

6.3.3.2. Test case funtion Login

Test Case ID	LG_001	**Test Case Description**	Test login functionality.		
Created By	Lam	**Reviewed By**	Lam	**Version**	1.1

QA Tester's Log	Review comments from Lam incorprate in version 1.0

Tester's Name	Lam	**Date Tested**	July 1, 2020	**Test Case (Pass/Fail/Not Executed)**	Pass

S #	Prerequisites:
1	Visit Google Chrome
2	Go to Url: http: // localhost: 4200
3	On the Home page, display the Login Form on the NavBar
4	

S #	Test Data
1	Username = "tommy"
2	Password = "Password1"
3	
4	

Test Scenario	Verify that the entered data is valid, and the user successfully logged on to the system. And have the right permissions according to your account.

Step #	Step Details	Expected Results	Actual Results	Pass / Fail / Not executed / Suspended
1	Fill in the corresponding values in the Form: Username, Password.	Information is entered and displayed correctly. Password information is hidden.	As Expected	Pass
2	Click button Login.	Verify correct account information. Forward to home page. Currently, the user is logged into the system and can use the functions according to the system's authority.	As Expected	Pass
3				

4				

CHAPTER 7. CONCLUSION AND DEVELOPMENT STRATEGY

7.1. Results

Firstly, we worked, researched, practiced, understand, and grasped the advantages and disadvantages of some popular technologies, including ASP.NET Core, Angular 9, SQLite.

Thereby we have a knowledge about the processes of communication, authentication and authorization, the scale and complexity of a large system as well as the security, development, and expansion between services, the client, and the server via RESTful APIs and JWT for a commercial project.

We have a piece of knowledge to building the user application interface from the core knowledge researched and applied is the Angular framework, one of the quite popular frameworks now for building applications in the form of SPA - Single Page Application under the client-side rendering, help UI don’t need to reload the interface. Besides that, search, calculate and validate data from the user side to improve time, increase user interaction, and reduce the load on the server.

In addition, Angular helps increase software development performance through “Binding Data”, “Directive” as well as build by dividing the website into separate components and individual services, increasing the reuse and helping the interface of a website is unified and has its own characteristics.

7.2. Advantages

Microservices is a new architecture, bringing many advantages for large systems, many benefits of rapid development and expansion, meeting the availability of e-commerce system.

ASP.NET Core is a technology developed and maintained by Microsoft, which is now expanded to many different platforms and popular for many large systems, meeting the requirements of high load and security.

Angular is open sources developed by Google and Facebook which are currently a strong and popular framework now, bringing a good experience for users both on the website and mobile platforms.

7.3. Disadvantages

Microservices is a multi-service architecture, which requires a high-load environment for running multiple services at the same time.

The user interface is not really good due to the lack of experience in developing products for real users.

Transaction management for asynchronous processing operations, data flow management when making requests for many services is not really good, not ensuring the integrity of information.

7.4. Development strategy

- Develop customer authentication email and phone numbers to help verify customer information.

- Extending and absorbing comments from users helps to bring a better experience to the product interface.

- Completing the interface, fixing system errors, and rising security to bring up the testing environment and competing with available products.

- Function Development: Grammar Learning, Grammar Practice, Toeic Exam.

- Develop English or Vietnamese user voice recognition function and translate into Vietnamese or English languages.

REFERENCES

8.1. Vietnamese

[1] English learning website Tflat. Retrieved from: https://tienganhtflat.com/

[2] Duolingo English learning website. Retrieved from: https://www.duolingo.com/

[3] English learning website "Tiếng Anh Mỗi Ngày".

8.2. English

[1] Create web APIs with ASP.NET Core. Retrieved from: https://docs.microsoft.com/en-us/aspnet/core/web-api/?view=aspnetcore-3.1

[2] Angular Tour of Heroes app and tutorial. Retrieved from: https://angular.io/tutorial

[3] Top 10 Advantages of .NET Core. Retrieved from: https://www.fortech.ro/top-advantages-net-core/

[4] Top 13 ASP.NET Core Features You Need to Know. Retrieved from: https://stackify.com/asp-net-core-features/

[5] Advantages and Disadvantages of Angular. Retrieved from: https://www.knowledgehut.com/blog/web-development/advantages-and-disadvantages-of-angular

[6] Top 10 Features of Angular 9. Retrieved from: https://medium.com/@angularminds/top-10-features-of-angular-9-dd9f067ddee1