

Ha Ngoc Linh

Supervisor: Teacher Do Tien Thanh

November 2020

RubySmile Text-to-speech System

Word count: 16455 words

A dissertation submitted in partial fulfilment of the University of
Greenwich's BSc Computing

Abstract

People are living in the age of technology. With the development of artificial intelligence (AI), machines can no longer only communicate with humans through text messages, they can also communicate with sounds, emitting voice messages. Machine voice is more and more developed closer to human voices. Big companies like Google and Amazon have researched this technology and put it into practice for use. However, no matter how much it is developed, the voice still lacks one thing that is emotion.

Besides the development of technology, new human needs also appeared. Station announcements, airports, story-telling videos, reading books, voice-overs, ... these applications are now moving from real human voices to mechanical voices. The only problem with this conversion is that there is no emotion in the machine voice. Same sentence but hundreds of emotions with many different readings. And RubySmile Text2Speech application will be born to solve that problem.

This report will include searching of similar products, requirement analysis, market analysis. This document will also present the process of product design and development, giving a future plan to turn RubySmile Text2Speech into a practical application known to everyone.

Preface

The market demand for audio products is huge. In terms of audiobooks alone, the market size in 2019 is valued at \$ 2.67 billion and is expected to increase sharply from 2020 to 2027 with an annual increase of 24.4%. Not to mention the demand for more narrative films, the market for the technology is huge and it's just in the early stages of repairing phase.

Therefore, this is a great time to research and develop a technology to meet the needs of users. However, because it is a new application with new features, the version in this report will only be a version with the purpose of collecting user feedback from that evaluating and improving the design and build a perfect application in the future.

The essence of the emotion in words is the fastness, slowness in speech and the emphasis in words. Applying this characteristics, the system will apply available Text to speech technologies from big companies such as Google, Amazon, FPT AI. Using SSML (Speech Synthesis Markup Language) allow users to customize their voice according to their wishes to create sound files as desired.

Acknowledgements

Many thanks to Mr. Do Tien Thanh for helping in completing the product and this report.

Contents

Abstract	2
Preface.....	3
Acknowledgements.....	4
I. Introduction	10
II. Application overview	10
A. Application Domain	10
1. Text-to-speech Definition	10
2. Current and trends	12
3. Application Purpose	16
B. Development methodologies	17
1. Waterfall	17
III. Agile	20
2. Suitable method for Text2Speech system.....	22
B. Technologies	23
1. Architectural.....	23
2. Frontend & Backend technology.....	28
3. Frontend	29
4. Backend.....	35
5. Speech Synthesis Markup Language (SSML).....	39
IV. Requirement analysis and Plan for RubySmile Text2Speech System	40
A. Review of other similar applications	40
1. Balabolka.....	40
2. Panopreter Basic	41
3. Analysis of applications in the market.....	42
B. Vocal Analysis.....	43
C. Limitations of current Text-to-speech technology and SSML solution	45
1. Limitations of current Text-to-speech.....	45
D. User requirement definition.....	46
E. Summary and Future Plan	50
V. Design of RubySmile Text2Speech System	51

A.	Architecture Diagram	51
B.	Usecase	52
C.	Sequence Diagram.....	53
D.	Wireframe	54
1.	Text-to-speech page	54
2.	Feedback page.....	55
3.	HowToUse and AboutUs page	56
VI.	Development of RubySmile Text2Speech System	57
A.	Frontend	57
1.	The idea of manipulation on the interface	57
2.	Structure	57
3.	Code flow	62
B.	Backend.....	70
1.	Structure	70
2.	Code flow	70
VII.	Testing.....	71
A.	Test plan.....	71
B.	Test case.....	71
C.	Summary	73
VIII.	Evaluation	74
A.	Human Interaction	74
1.	Visibility of system status	74
2.	Match between system and the real world.....	74
3.	User control and freedom	74
4.	Consistency and standards	74
5.	Error prevention	74
6.	Recognition rather than recall	75
7.	Flexibility and efficiency of use	75
8.	Aesthetic and minimalist design	75
9.	Help users recognize, diagnose, and recover from errors	75
10.	Help and documentation	75

B. Product review	76
C. Plan analysis	76
IX. Conclusion	77
A. Analysis skill	77
B. Development skill	77
C. Report writing skill.....	77
X. References	78
XI. Appendix A – Source code	81
XII. Survey result.....	81
XIII. Appendix C – Project Schedule	84
XIV. Appendix D – Screen Captures.....	85
A. Text to speech screen	85
B. Feedback screen.....	86
C. HowToUse screen.....	86
D. About us screen	87

Table of Figures

Figure 1: Text-to-speech (MobileScout, 2016)	10
Figure 2: Waterfall method	18
Figure 3: Overview Agile method (Bauer, 2019)	20
Figure 4: Agile method circle (Patel, 2019)	21
Figure 5: Microservice example (Kappagantula, 2019)	26
Figure 6: Angular vs React vs Vuejs (Sumon, 2020)	29
Figure 7: Popularity of Vue, React and Angular on Github (Cavalcanti, 2020)	33
Figure 8: IT developers' opinion (Cavalcanti, 2020)	34
Figure 9: Nodejs vs Laravel (Geekboots, 2020)	35
Figure 10: Interface of Balabolka Application	41
Figure 11: Interface of Panopreter Basic Application	42
Figure 12: Hello and Hellooo sound wave (Hildebrand, 2020)	43
Figure 13: A Four-Dimensional Conceptual Framework of Speech	44
Figure 14: Architecture diagram of RubySmile	51
Figure 15: sequence diagram	53
Figure 16: Wireframe Text-to-speech page	54
Figure 17: Wireframe Feedback page	55
Figure 18: Wireframe "How to use" page	56
Figure 19: Wireframe "About us" page	56
Figure 20: Source code structure	57
Figure 21: source code of pages folder	58
Figure 22: source code of components folder	59
Figure 23: source code of asset folder	60
Figure 24: source code of router folder	60
Figure 25: source code of plugins folder	61
Figure 26: source code of store folder	61
Figure 27: text-to-speech feature code flow	67
Figure 28: Code Flow feedback feature	69
Figure 29: structure code of backend	70
Figure 30: Have you ever had a need for voice recording at work or in life?	81
Figure 31: Have you ever had a need for voice recording at work or in life?	81
Figure 32: How much is your need to use these voices?	81
Figure 33: Do you know some tools that use text to speech technology?	82
Figure 34: Have you ever tried using tools to get voices for your product?	82
Figure 35: Are the tools that you have been using for fees?	82
Figure 36: What is the first thing of a good application you will notice?	83
Figure 37: If it's free, would you be willing to change from voice recording to using text-to-speech technology?	83
Figure 38: Project plan	84

Figure 39: Homepage screen	85
Figure 40: Feedback screen	86
Figure 41: HowToUse Screen.....	86
Figure 42: About us screen	87

I. Introduction

This section is about Text-To-Speech technology domain and how it effects in our life. Accordingly, the potentials of this technology will be determined. Through this section, the reader will be able to understand why it is necessary to build such an application in life. In addition, this section will also show the challenges that will be faced. From there, we can have the most suitable solution.

The last but not least, there are the methods and technologies that can be used to build the app. Different technologies will be identified and compared to choose the most suitable technology.

From there, the application will be able to bring a better experience for users, their intended use.

II. Application overview

The application focuses on the normal users, who absolutely have no knowledge of technology. This product applies text-to-speech technology to give users the ability to customize their audio output.

For people wishing to convert text to speech. Instead of using audio file editing and merging applications, users can directly create the desired audio file from the input text.

A. Application Domain

1. Text-to-speech Definition

a) Text-to-speech (TTS)



Figure 1: Text-to-speech (MobileScout, 2016)

Based on (Taylor, Text-to-speech synthesis, 2009) Text to speech is a natural language and text understanding technology based on artificial intelligence to create a complete synthesized sound with consistent rhythm and intonation.

Text to speech, which has been studied for several decades around the world, has been particularly strong in the last 10 to 15 years. This technology has been developed almost completely abroad, especially in English-speaking countries, bringing many practical values in business and life.

With Text-To-Speech (TTS) technology, robot communication becomes easier and more natural than ever. Text-To-Speech can be used in automated answering switchboard systems, public notification systems, virtual assistants, audio newspapers, audio books, film narrations, and so on.

b) Speech Synthesis Markup Language (SSML)

According to (Isard, 1995)

The use of text as input to the voice system is attractive from the human user's point of view, as it means the user doesn't have to do any action with the text to convert it from the form which can be read by human into machine-readable. However, text is not the ideal input from a machine standpoint, as the process of translating text to speech is complicated. In a TTS system, the document has to go through several stages before it is ready to be compiled. The process is computationally costly and error prone, as human readable text is both ambiguous and lacks specific in terms of linguistic information, nor can machines understand emotions in text entered.

TTS systems use a variety of methods to try to solve these problems, however, all of these methods are system specification, so no standard solution has emerged.

Speech-synthesized markup language (SSML) provides a standard method for dealing with the problems outlined above.

In SSML, markers are added to the text that contains additional information that the text itself does not specify. SSML is deployed as an application of the Standard General Markup Language

(SGML). In essence a set of rules is defined by a system to specify what can make up an SSML document, and the documents can be written according to these rules. In the SSML version presented in this thesis, for example, it is possible to highlight the beginning and the end of each phrase, and clearly mark which word in the sentence needs to be emphasized. Some of the information contained in an SSML tag can be extracted from text using traditional TTS methods, but SSML also allows to include quality designating tags such as speech pitch or volume, or the mood of the text.

To make speech processing easy, we need to rely on parameters, so SSML was born to create a common standard that governs parameters.

Developed by the W3C, nowadays, SSML is one of the standards applied in speech synthesis systems, and the language is built on an XML platform, which helps to parse and process the data easier.

Since SSML is based on XML, it must follow the structure and syntax of XML, so the SSML namespace must be clearly defined, and <http://www.w3.org/2001/10/synthesis> is the standard SSML namespace.

2. Current and trends

a) Benefits of Text to Speech Software

(Text to speech) also known as the technology "converting text to speech" is the era of words synthesized from text. This technology is used to communicate or communicate with users when they are unable to read the content on the screen or reading too much text is quite inconvenient.

This not only opens up new uses for mobile apps and information, but also has the potential to turn the world into a more accessible place for people who can't / are lazy to read text on the screen.

The technology behind converting text to speech has evolved over the past few decades. Using deep research, text transitions can now produce very natural-sounding speech that includes changes in pitch, rate, pronunciation, and speech variations. Today, computer generated speech is used in a variety of situations and is slowly becoming a common factor in user interface

building. Reading news, gaming, public notification systems, e-learning, phone protocol, IoT devices & applications, and virtual personal assistants are just a few of the starting points of the type of literature transformation technology. this voice version.

Voice synthesis makes applications more accessible, allowing people to consume and understand information without having to focus on the screen.

Applications that use voice to communicate are becoming more common every day. With text-to-speech solutions, web pages, mobile apps, e-books, learning tools, and online documents can have their own say in their own right.

Audio publishing:

Publishers and content owners can quickly and inexpensively convert books, articles, and any written document into audio with text-to-speech technology.

Online learning and training:

Text-to-speech converter provides an easy way to convert learning content into a format that is both more efficient and less expensive and widely deployed in multiple languages.

Customer service:

With the use of natural voice, text-to-speech conversion can enhance the PBX's interactive quality and support communication applications.

Media & Entertainment:

When it comes to operating sound generation, text-to-speech technology can also help reduce costs and increase efficiency in pre-production and product development.

b) Trending

(1) Top text to speech software

Firstly, to understand the trend of this technology, we must know the top Best text to speech software of 2020. According to website (Nicholas Fearn, 2020), there are some text-to-speech application:

- Amazon Polly - A voice synthesis solution for developers. Using advanced deep learning techniques, the software turns text into speech lifelike as real people speak. Developers can use this software to create voice-enabled products and applications. It has an API that allows you to easily integrate voice synthesis into e-books, articles, and other media. The great thing is that Polly is very easy to use. To convert text to speech, you simply send it via the API and it will send a stream of audio back to your app. You can also store audio streams as MP3, Vorbis and PCM files, and there is support for many international languages and dialects. These include British British, American British, Australian British, French, German, Italian, Spanish, Dutch, Danish and Russian.

- Linguattec Voice Reader - A reliable text-to-speech application. With this software, you can convert text such as Word documents, emails, EPUBs and PDF documents into audio streams quickly. You can then listen to them on your PC or mobile device. You can choose from 67 different voices and support up to 45 languages such as voices. The purpose of this software is to improve productivity. For example, you can download an app to read manuscripts for speeches, lectures, or presentations to find out incorrect word order or omitted words.

- Capti Voice - This application is developed to cater to the education industry. Capti Voice lets you listen to whatever you want to read. With it, you can personalize learning and teaching, and overcome language barriers. Capti Voice is used by many schools, colleges, businesses and professionals worldwide. Supporting more than 20 languages, the app can be used to improve vocabulary. It can report a variety of content, including e-books, articles and web pages.

- Natural Reader - This software is enhanced by cloud-based technology. For more personal use, the solution allows you to convert written text such as Word and PDF documents, e-books and web pages into human-like speech.

- Voice Dream Reader - An option optimized for mobile devices. Voice Dream Reader can convert documents, web articles and e-books into natural sounding voices. The app comes with 186 built-in voices in over 30 languages. You can download the software to read articles while driving, work, or exercise, and there's auto-scrolling, full-screen, and distraction-free modes to keep you

focused. Voice Dream Reader can be used with cloud solutions like Dropbox, Google Drive, iCloud Drive, Pocket, Instapaper and Evernote.

In addition, many softwares use text to speech technology such as:

- Balabolka
- Panopreter Basic
- WordTalk
- Zabaware Text-to-Speech Reader

(2) *Google trends*

Google Trends is a Google service that helps users compare Google Search search results globally by giving statistical results, comparing keyword trends over time in each country or above. the whole world.

Google Trends will display data in the form of a chart over time to help users easily identify when the keyword's interest level increases, when there are signs of "cooling" from there. changes in line with new consumer trends.

Using Google Trends with keyword “text to speech” we obtained the following results



This is the result of the trend of searching for the phrase "text to speech" in the last 5 years (from October 2015 to October 2020). Values above 50 represent above average interest levels. This phrase is getting more and more attention recently.

In addition, when searching the phrase "text to speech" on google, we can immediately receive 831000 search results in just 0.68 seconds. This proves the high level of interest in this phrase.

(3) *Evaluation*

We can see text to speech as a growing market. More and more people are interested in this field. The demand is also gradually increasing, reflected in the increasing keyword "text to speech". Even this keyword itself has always had a fairly high level of interest in the last 5 years.

Currently the application of text to speech is quite focused on the programmers. Applications are built primarily in the form of an API so that developers can integrate them into their applications.

However, there are also user-centric applications. However, the number of applications is still very limited. Most applications translate text to speech directly and offer very little customizability for the user.

3. Application Purpose

Life is developing day by day, people become more and more hasty, the faster pace of life comes with the need for more information updates. People do not have time to read books, but they still want to know the content of interesting books. From these needs, on social networking sites and video platforms, a new trend is emerging in reading books and storytelling videos. Videos of storytelling continuously have reached the top trending on famous platforms such as Youtube, Kwai, Tiktok, ... More and more reading applications are also increasing, many applications are in the top of app markets. However, the common point of this market is using human voice as the product. Reading costs are high, recording takes a lot of time, editing post-recording time is also expensive, productivity is not high, and there are many risks when hiring readers. Because of these, many companies have opted to Text-to-speech technology. This reduces the cost of hiring readers, reducing recording time, and reducing human resources. Modern technology, machine voice is now more and more human, making the product better and better. However, no matter how much it develops, the machine voice is difficult to bring emotions such as human voice, there are sentences and passages that require changing the tone, current technology is difficult to do it. In addition, using text-to-speech technology makes it easy to make a product simple with just a single voice.

From the above characteristics, the idea of RubySmile application was born. The application will focus on increasing customizability for the user. The application will focus on the end user. The

application will give them the ability to adjust the tone, change voice in a flexible, customizable way. Users will be able to easily listen to and download their results to the computer.

The application will use existing text-to-speech technologies, applying SSML to give the user customization capabilities. It will simplify and visualize the using process so that the user can easily understand and use it.

B. Development methodologies

1. Waterfall

Based on (Bassil, 2012)

Waterfall model was the first SDLC (Software Development Life Cycle) introduced in the world. It is also known as the linear sequential life cycle model. Its biggest advantage is that it is simple to understand and use. In the waterfall model, each stage had to be completed before the next could take place. Everything that happened in the waterfall model had to be in sequence. This software development model is usually developed for small projects and does not have high requirements. At the end of each phase, the development team will review to determine if the project is on track and on schedule, and then make adjustments in the next phase. In this model, the certification process is started after the product is completed. In the waterfall model, the phases do not overlap.

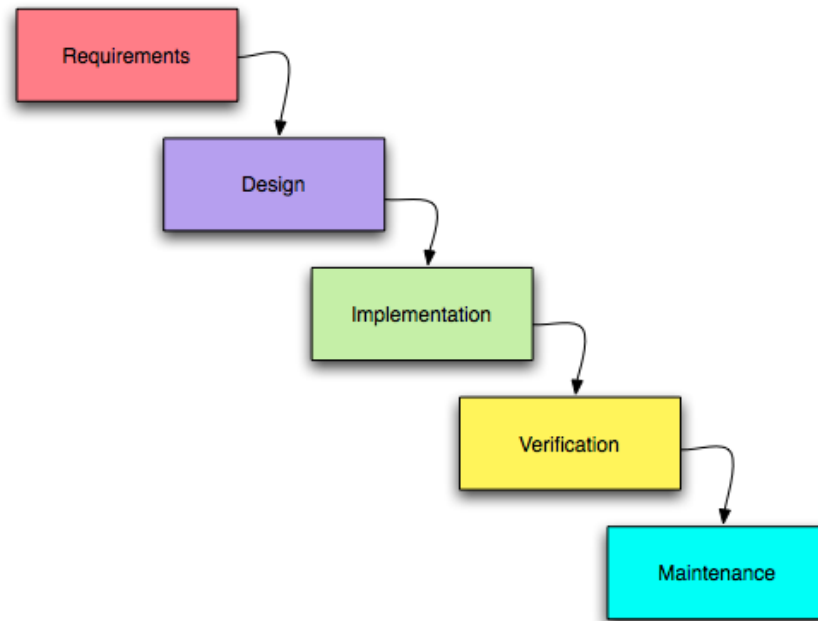


Figure 2: Waterfall method

According to (Hughey, 2009), there are some advantages and disadvantages:

a) Advantages of waterfall model

- *Use clear structure*

Comparing the waterfall method with other methodologies, Waterfall focused heavily on a series of clearly defined stages. Its structure is quite simple:

- Requirement analysis
- System design, UI design
- Implementation
- Testing, verification
- Deployment and Maintenance

The development team following this method must complete one step before moving on to the next, so if there are any problems to complete a step, they will be presented with an easy way to analyze. Unlike the other method (Agile, Scrum) Waterfall does not require certification or training because of its simplicity. For a person who has never heard of Waterfall, if this method is sketched in the form of a chart, that person can easily understand and follow that method.

- *Determine the end goal early*

One of the advantages of Waterfall is that the objectives of each step is clearly defined, the final goal of the project is also clearly outlined. For small projects, this goal defining step, that makes the development team aware of the overall goal from the start, less likely to deviate from the project point while the project is in the process of running.

Unlike Scrum, which divides a project into many sprints, Waterfall always focuses on the end goal. If the development team has a specific goal and a clear deadline, the Waterfall method will help eliminate the risk of deviating from that goal.

- *Good information transmission*

Waterfall methodology focuses on clear information transfer at each step. During the project cycle, the development team will record information for each stage. No matter what step the development team left behind or encountered an unexpected change, thanks to the information conveyed in each stage of the Waterfall method, members can easily catch up with speed.

b) Disadvantages of waterfall method

- *Make changes difficult*

Waterfall method goes in a straight line, and the objectives are clearly defined. Implementing this method, the development team cannot go back through completed steps. Basically, there's barely any room for unexpected changes or modifications. If your team has been sticking to Waterfall's rules as it gets closer to the end of the project, if changes occur that require changing the scope or objectives of the project, it will not be easy. A sudden change to any part of a project could make most of the work become useless, maybe loss all the work.

- *Excluding customers and /or end users*

As an internal process, Waterfall methodology has almost no concern for the end user or the client participating in the project. Its main purpose is helping the development team carry out the phases of a project internally. However, in reality, clients often want to participate directly in the project, they want to comment and sometimes modify a function, a feature or modify project requirements.

- *Delay testing until completion*

Delaying testing to the end of the project is a huge risk, but Waterfall is required to take all the steps to complete the project in order to do this. In a large project, a product can consist of many parts, one part will have links with the rest, a mistake in one part can lead to all mistakes for the other. The project can take a long time to complete, and when testing detects errors, with a large-scale project, modifications can cause significant delays.

c) Summary

The waterfall model forces the development team to follow a certain sequence of stages and never moves forward until the previous stage is completed. Basically, there's barely any room for unexpected changes or modifications. If the development team is small and the project is small too and the changes are predictable, then Waterfall might be an ideal method.

III. Agile

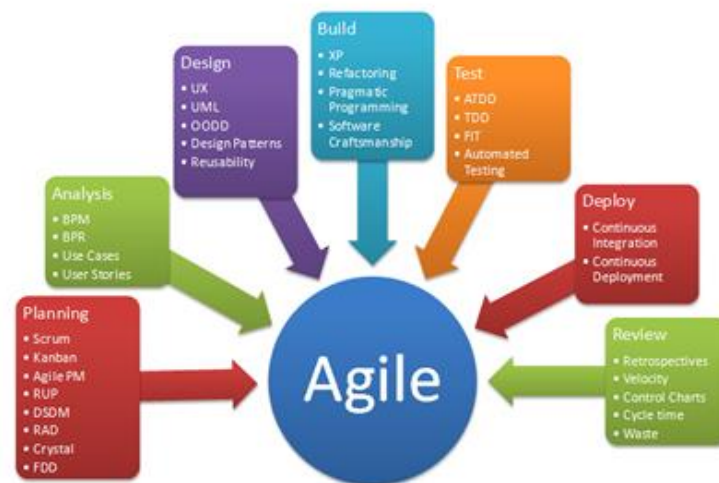


Figure 3: Overview Agile method (Bauer, 2019)

Based on (Pekka Abrahamsson, 2002)

Agile is an SDLC method where a development team, when applying it, will be able to manage a project by dividing it into different phases, the development process will have continuous collaboration with stakeholders and improvement. progressing in stages, these processes will be

repeated in every stage. Applying the Agile method, the development team will start with the customer describing the final product, giving the requirements. The client will clarify their expectations for the project. The development team will then rotate to the planning, implementation and evaluation process - a process that can be altered to accommodate both parties. Collaboration and constant change are the two most important things about this approach.

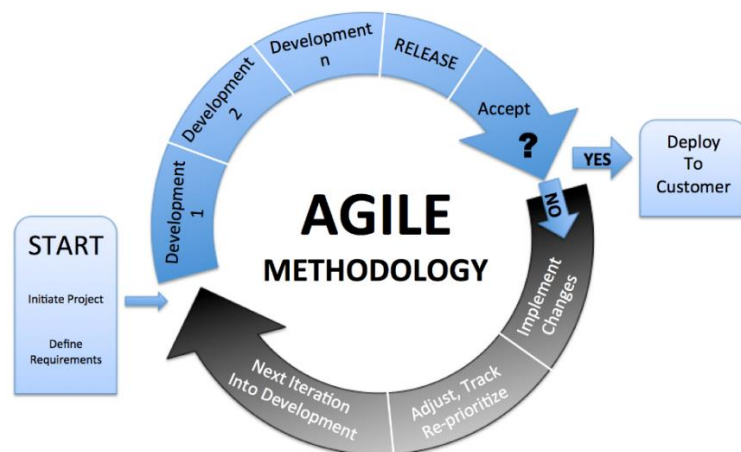


Figure 4: Agile method circle (Patel, 2019)

According to (Koch, 2004) There are some advantages and disadvantages of this method

a) Advantages of Agile method

- Using Agile will increase the relationship with customers, helping customers to always be satisfied because the product will be distributed quickly and continuously updated in stages.
- Products and work results are distributed regularly (weekly instead of months).
- Tight, everyday cooperation between the customer and the development team, this helps the requirement will always be adhered and each change will avoid too much change.
- Design and technique will be changed easily to suit the project situation after each stage.
- Adapt well to changing circumstances is the most prominent characteristics of Agile.
- Unlike Waterfall, even late changes in project requirements don't become complicated.

b) Disadvantage of Agile method

- Since using Agile makes it easy to change the requirement, the project can easily deviate from the original assumption if the customer is not sure what they want the final outcome to be.
- Agile requires the development team to have experience and knowledge of this method. Only advanced programmers are capable of making decisions during development. Hence, it has no room for new programmers unless combined with experienced resources.

c) Summary

- As new changes are constantly being introduced and need a combination of the development team and the client side, Agile will be the perfect choice. The freedom to change any detail during the project is very important. New changes can be made at a very low cost because the requirement is constantly being exchanged with the customer, continuous corrections are applied minimizing the magnitude of each change, leading to cost reduction and reduction. development time.
- Unlike the waterfall model, it is necessary to limit the number of planning times in the project. Agile always assumes that the end user will always change their request. Changes can be discussed and features can be implemented new or removed based on feedback. This is effective for the customer and the complete system will be more complete as they want or need.

2. Suitable method for Text2Speech system

With a far-off plan for the project, Text2Speech brings a completely new experience, Text2Speech application with the goal of being an application never before in the world. The product will need to collect a lot of feedback in order to revise and upgrade. Application will take a lot of time to develop and has to go through many tests. Because of these characteristics, it is easy to see that the suitable method for this project is Agile. But for only this version of project, because the period is short time, development team has only 1 developer, there is no other choice except Waterfall. So for only this current version of this project, development methodology is Waterfall.

B. Technologies

1. Architectural

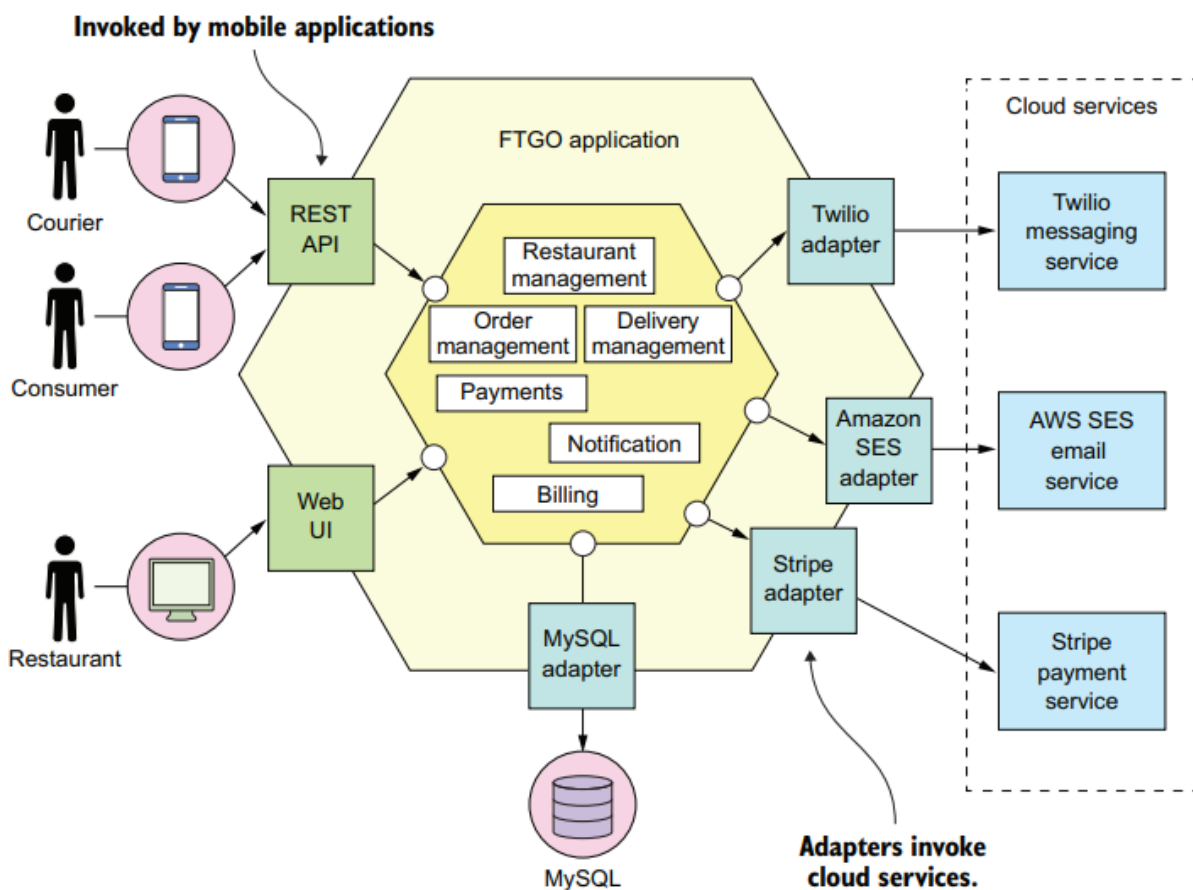
On the basis of (Newman, 2020), When developing an application you can choose to develop that application with a hexagonal architecture or a layered architecture that includes different types of components:

- Presentation - responsible for handling HTTP requests and responses using HTML or JSON / XML.
- Business logic - the business logic of the application.
- Access to the database - the data access objects are responsible for accessing the database.
- Application integration - integration with other services.

a) Monolithic

(RICHARDSON,

2019)



(1) *Advantages*

There are several advantages when applying monolithic software architecture in building systems including:

- Easy to develop
- Easy to make changes in code in the future.
- Easy for testing, easy to implement, easy for scaling.

There is a prime example of the benefits of a monolithic software architecture, during project implementation, the development team only needs to focus on a few IDEs (Integrated Development Environment) and certain tools to build the whole system suite instead of lots of different IDEs and loads of other tools. In fact, there are many IDEs that we must pay for them, so instead of using multiple IDEs, the development team only needs to use one or two IDEs to build entire system. Therefore, product development costs will decrease.

In fact, the application will be changed over time, it will be updated and modified a lot. Using monolithic software architecture, these changes will be easier to do. For example, if you want to change the database, the development team only needs to make modifications on the database with small changes in a single system, this will be much easier than having to edit in many different systems with complex interlinked data streams.

The testing process plays a very important role in the project implementation. Risks can arise if this process is not done seriously and carefully. The monolithic software architecture allows this process to be done easily. For example, if testing is done at the interface and detects an error in the frontend response data, the development team will easily go backwards to test the APIs. Since these APIs are all written in a single folder, finding errors is quick and easy. Also, since there is only 1 project file, the system implementation only has to be done once and the configuration is not too much and takes time. In addition, the ease of scaling a system using monolithic software architecture is undeniable. Because of the load balancer mechanism, the system expansion is done easily.

It is true that monolithic architecture has been used with great success for the past few decades. And many successful famous applications have been built according to monolithic software

architecture. Many applications of large-scale enterprises are still being deployed under monolithic architecture. All thanks to its most outstanding feature is the ease and low cost. However, the world has changed, technology has changed, and software requirements have also changed, all of these things have revealed many weaknesses of monolithic architecture.

(2) Disadvantages

Flexibility: The first thing to criticize about monolithic architecture is its inflexibility. Monolithic architecture cannot use many different technologies. The project's technological foundation is decided from the very beginning of the project and follows through to the end of the project. Once the development is at a provisional level, upgrading will immediately become difficult, not to mention the adoption of another new technology.

Reliability: The truth is, this structure is not reliable. Just one feature crashes, the entire app can stop working.

Development Speed: The speed of development of an application using a monolithic structure is really slow. It will be difficult for members to work together in the same project file. The quality of the project's code will decrease over time. In addition to the increasing project size, the IDE will gradually become overloaded, the machines will become slower. The larger the application capacity, the longer the application startup time. All these factors will have a huge impact on the work efficiency of the development team and the final product.

Building complex applications: Using multiple technologies together simplifies the development of complex applications, but monolithic architecture can only use a very limited amount of work. turmeric. This made the development process much more difficult.

Scalability: It's true that monolithic architecture-built applications are easy to scale, but when they reach a certain magnitude, scaling becomes more and more complex than ever. In addition, an application grows in size with many components of which may consume a lot of CPU while another component can consume a lot of memory. With monolithic architecture, it is impossible to scale each component independently.

Continuous deployment: As the size of the application grows, continuous deployment becomes extremely difficult. The size of the application is the biggest obstacle. Only changing a small component, but we will have to re-deploy the large system, which takes time and effort.

b) Microservices

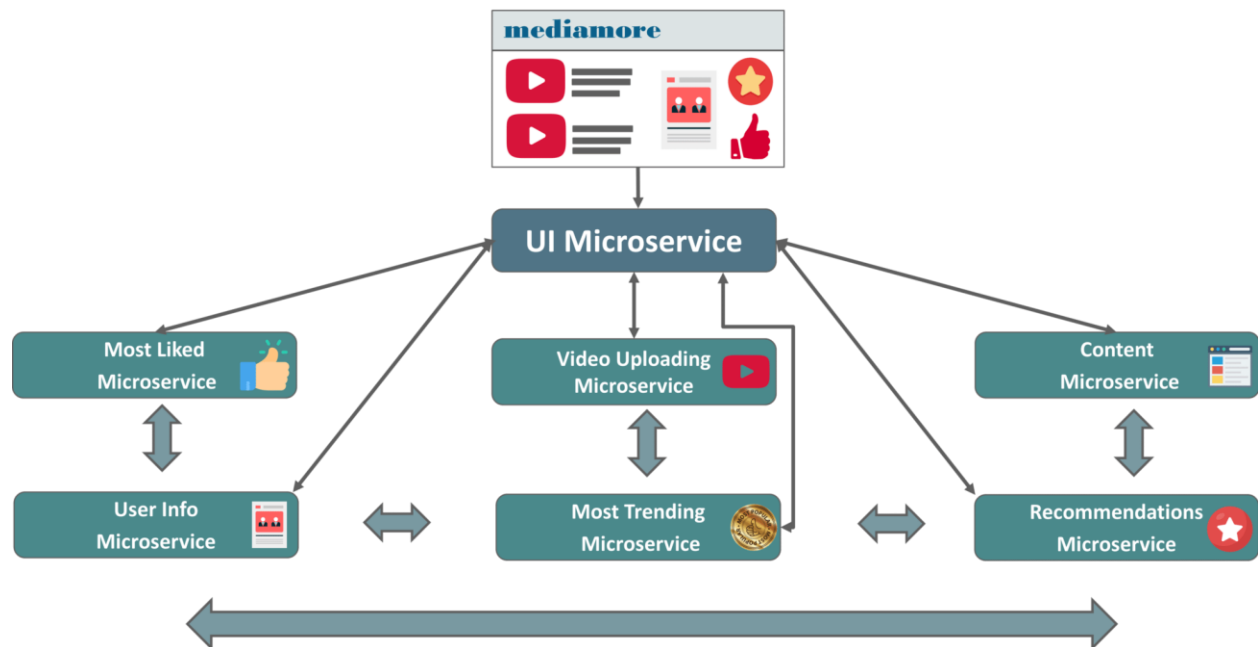


Figure 5: Microservice example (Kappagantula, 2019)

(1) Advantages

There are many advantages to using microservices:

- The complexity of the system will be reduced due to the segregation of services for a specific object. Therefore, if there is a new developer joining the project, that developer can easily participate in the development process without understanding too much of the entire system.
- The development team can easily use many different platforms and technologies in 1 project. This optimizes the power of each technology for a different job. Functionality will be optimized in the best way.

- Small services, easy to maintain and deploy. Each service in the microservices architecture can be understood as a subsystem. Therefore, the development team can deploy each service independently. This will result in each service being independently scalable.

-

(2) *Disadvantages*

There are a few downsides to microservices:

- Minimizing services is not always the right choice. A service can be a too small service or it can be a too large service. In both of these cases, service splitting is a big problem. To get an exact decision requires careful analysis. This requires more calculation than monolithic.
- Unlike monolithic applications, testing in microservice architecture is not easy because if the developers want to test one service, the developers may have to run another service, since the services have interconnection.
- Implementing microservices is not an easy task. Since each service may have to run in a different environment, the team will have to do a lot of configuring. Therefore, the development team needs to use some technologies to implement microservices such as Docker, Kubernetes, ...

c) *The most suitable architectural*

After making the comparison, making the judgments between the two monolithic architectures and microservices, both architectures have specific aspects for the project. Initially, after evaluation, microservices seems to be a perfect choice for the project, because of the ability to easily expand, upgrade, process speed, response speed it brings. However, in the test version of this project, monolithic should be chosen. Because of the short development time, the familiarity of this architecture to developers and servers is not too complicated. Therefore, Monolithic will be the option for this project.

2. Frontend & Backend technology

Frontend and Backend are two opposite terms in web development work. Each party has a separate function and needs to communicate and exchange information with the other to complete the functionality of the website.

Frontend development: The part of a website that users see and interact directly with is called the user interface (also known as the client side). The job of creating that main interface and frontend development. The creation process includes user interface and user experience: color and style of text, images, graphs and tables, buttons, colors and navigation menus ... interactive experiences ... HTML, CSS and Javascript are languages used for Front End development. Nowadays, with the use of those languages, to make frontend development easier, programmers have created frameworks and libraries such as bootstrap to increase the aesthetics of the interface, frameworks (Vuejs, React, Angular ...) to reduce development complexity and speed up the development process

Backend development: The backend refers to the parts that are located on the server side of a web page. It stores and organizes data, and ensures everything on the client side of the website works well. If the frontend is something that the user perceives (seeing, interaction) then the backend is a part that you cannot see and interact with. Here, although the user cannot interact directly, it is the place where most of the system's computational logic, they indirectly interact with the user. The backend work includes writing APIs, creating database, creating libraries, and working with system components, etc.

Before I go into 2 next sections, to avoid wasting time, I will only analyze technologies based on three factors:

- Advantages and disadvantages
- Popularity
- Understanding with this technology

This project is developed with a development participant consisting of only 1 person. Since development time is also limited, a deep understanding technology takes precedence. Then, I

will evaluate the popularity and ultimately the advantages and disadvantages of the technology with the project. Since the frontend technologies are highly developed currently, for the leading technologies, their capabilities are nearly the same. By evaluating this priority, I will be able to shorten project implementation time. I will analysis 3 frontend technologies: Angular, Reactjs, VuejsBy. With backend, I will evaluate Laravel and Nodejs.

3. Frontend



Figure 6: Angular vs React vs Vuejs (Sumon, 2020)

a) Angular

(1) Definition

According to the official Angular documentation (Angular), Angular is a Front-End framework supported by Google. Angular was first introduced in 2009. Then in 2016, Angular 2 was launched with many changes that bring out the superior. Angular can be used to develop mobile, web and even desktop application interfaces. Angular helps develop SPA (Single-page Application). This is a browser-based application that is not required to reload the page when using it.

(2) *Advantages and Disadvantages*

Advantages:

- TypeScript Support: Angular was created to be used with TypeScript. And TypeScript has the advantages of both Js and OOP languages.
- Data-binding: It automatically synchronizes data between the model element and the view.
- Detailed and complete documentation: This is a huge plus for Angular. With detailed documentation from the supplier itself, it will help newbies to learn Angular quickly access and master this framework. Thanks to that, reducing training time, discussing with colleagues ... when on their homepage there is all.
- MVVM (Model-View-ViewModel): allows a developer to divide the project into several independent parts. From there, programmers will easily edit, add features and maintain the project later.
- Dependency Injection: This is probably the feature that developers like most in AngularJS. DI is a design pattern that is mentioned a lot in the SOLID clean code philosophy. With DI, it allows developers to reduce dependencies between Objects. Objects are as independent as possible, the purpose of later upgrading, modifying... will limit the impact on other objects.
- Angular's architecture and architecture is specially created for large project scalability.

Disadvantage

- Many times the diversity and abundance are the downside. Angular is quite diverse in components / concepts such as Injectables, Components, Pipes, Modules ... This makes it a little more difficult to learn than Reactjs, which has only one component is a component.
- On a performance review, Angular seems to be slower than React and Vuejs.

(3) *Usage and popularity*

Companies using Angular: Microsoft, Autodesk, MacDonald's, UPS, Cisco partners, AT&T, Apple, Adobe, GoPro, ProtonMail, Clarity Design System, Upwork, Freelancer, Udemy, YouTube, Paypal, Nike, Google, Google, Telegram, Weather, iStockphoto, AWS, Crunchbase ...

However, in terms of personal knowledge with Angular, I only stop at the level of understanding and use. My understanding can be rated at 3/10.

b) React

(1) Definition

React is a JavaScript library, which Facebook has been around since 2013. This is a great library for building large web applications where data can change frequently.

(2) Advantages and Disadvantages

Advantages:

- Easy to learn, thanks to a simple design, using JSX (a syntax like HTML) to create templates. Facebook documentation is very detailed.
- Application speed is extremely impressive. This is all thanks to React's virtual DOM technique and its rendering optimization.
- Support for server-side rendering is very good. This makes React a powerful framework for content-centric applications.
- Support creating Progressive Web App (PWA) quickly. It's as simple as the command: "creat-react-app".
- Data-binding is one-way, meaning less unwanted side effects.
- The Redux model: a very good model for application state management. And the cool thing about it is that it's easy to learn.
- React follows the Functional Programming school, which generates code that is easy to test and reusable for cases.

Disadvantages:

- Lack of mainstream documentation: Because of ReactJS's super-fast development, the master documentation can't keep up. Most of the tutorials on the internet are outdated. This causes many difficulties for their home developers.
- React is moving towards functional programming, which will be a bit annoying and a bit of an aversion to developers familiar with object-oriented programming (OOP).
- Mixing templating with application logic (JSX) can confuse some developers at first.

(3) Usage and popularity

Companies using React: Facebook, Instagram, Netflix, New York Times, Yahoo, Khan Academy, Whatsapp, Codecademy, Dropbox, Airbnb, Asana, Atlassian, intercom, Microsoft, Slack, Storybook ...

However, out of the 3 frameworks, my knowledge of React is the least. This can be difficult and time consuming if I choose this framework.

c) Vuejs

(1) Definition

Vue.js is a JavaScript framework, released in 2013, perfectly suited for creating highly adaptable user interfaces and complex single page applications.

(2) Advantages and Disadvantages

Advantages:

- Empowered HTML: this means that Vue.js has many similar characteristics to Angular. Therefore it can help optimize the handling of HTML blocks using different elements.
- Detailed documentation: Thanks to the detailed documentation, learning is fast, saving application development using only the basics of HTML and JavaScript.
- Compatibility. The transition to Vuejs from the Js framework is relatively quick. Because Vuejs is quite similar to Angular and React in terms of design and architecture.
- Good integration. Vuejs can be used for building both complex single-page applications or just part of an application. This allows you to update and upgrade applications without having too much impact on the current system.
- Large scaling: js can develop highly reusable templates.
- Small size. The entire js library is only 20KB in size. Very small and much smaller than the other 2 frameworks

Disadvantages:

- Lack of resources: Vuejs still has a relatively small market share compared to React or Angular. That means that knowledge sharing within the community is still quite small. However, in

terms of development speed, Vuejs is leading compared to the other 2 frameworks. In the future, Vuejs will not be inferior to React and Angular.

(3) *Usage and popularity*

Companies using Vue.js: Xiaomi, Alibaba, WizzAir, EuroNews, Grammarly, Gitlab and Laracasts, Adobe, Behance, Codeship, Reuters.

In terms of my understanding, Vuejs is better than the other two frameworks. I have over 1 year working experience with Vuejs.

d) *The most suitable frontend technology for RubySmile Application*

Firstly, to make decision which framework should be, let check the popularity of 3 frameworks on Github (the largest platform for hosting IT projects)

GitHub Stats

	stars 🌟	forks 🍴	issues ⚠️
angular	45,947	12,027	2,317
vue	123,024	18,553	182
react	123,934	22,493	421

Figure 7: Popularity of Vue, React and Angular on Github (Cavalcanti, 2020)



Figure 8: IT developers' opinion (Cavalcanti, 2020)

First, considering the pros and cons of all 3 frameworks, all 3 have their great advantages and are quite similar. However, due to its later birth, learning from previous frameworks, Vuejs was born with the advantages of both React and Angular frameworks.

Next in terms of popularity. This is quite important because programming must be based on the principles of community, that is to learn from each other to come up and solve problems. The fact that a framework has a large community means it is easier to use the framework. When having problems with the development process, the developer can more easily solve it if there is a large community. Of the three frameworks, React seems to have the largest community. Both React and Angular are sponsored by two giants in the tech industry, Google and Facebook.

Finally, the development team's understanding of that technology. As I said from the beginning, this will be my top priority. Although Vuejs is a new framework, the community is very small, but it is not difficult at all in finding help in the Vuejs community. Vuejs has also learned a lot from React and Angular along with being updated regularly compared to the other 2 frameworks. My choice for this project is Vuejs.

4. Backend



Figure 9: Nodejs vs Laravel (Geekboots, 2020)

Before entering this section, there are a few things that need to be corrected. Why Laravel and Nodejs?

It seems a bit weird, but Laravel is a PHP framework and Nodejs is a cross-platform built on the V8 JavaScript Engine - the interpreter that executes JavaScript code, Nodejs makes JavaScript usable for Backend programming. Framework and Platform are completely different. Therefore, to correct this section, the confusion will be Laravel and ExpressJs (a framework for Nodejs).

The reason I wonder between these 2 technologies is because these are the only 2 technologies I have knowledge of about them. It will be more advantageous if implementing a project using technology that you know.

a) Laravel

After research Laravel home page (Otwell), There are some advantages and disadvantages about it.

(1) Definition

Laravel is an open-source PHP framework, it is simple, easy to understand and easy to learn. It follows the design pattern MVC (Model-View-Controller). It is the best framework for PHP developers.

(2) Advantages and Disadvantages

Advantages:

- **Template Engine:** Laravel framework is appreciated for its lightweight templates built-in to use to create great layouts. Users just need to put the appropriate code content to be able to easily follow it. Not only does it provide stable PHP constructs, but it also provides some built-in CSS widgets and JS code even with frontend frameworks like React, Vuejs.
- **MVC architecture support:** Laravel follows the MVC architecture model, which allows separation between logic and interfaces. This MVC pattern comes with a number of built-in features that enhance application performance and improve both security and usability.
- **Security:** The Laravel framework provides very strong security for Web applications. It integrates many built-in security measures to ensure the convenience of use and high security.
- **Database migration system:** Laravel has built-in migration, which makes it convenient every time you reinstall the project, the project will be run again without wasting database creation time. Due to such functionality, the risk of data loss is extremely small. This not only provides the basis for changing the structure of the database, but also allows the use of PHP code instead of SQL.
- Laravel is supported by ORM, there are abstraction, automation.
- Laravel document is very clear to read, that make Laravel become very good choice.

Disadvantages:

- Laravel often needs third-party integrations to build custom websites (Lots of other libraries and frameworks included).
- Laravel is quite slow compared to other frameworks, slow in development and end product performance, and programmers need to have a good knowledge of PHP before they can use Laravel.

(3) Usage and popularity

Follow information on website (BuiltWith) There are 1,190,661 public website using Laravel. With keyword “Laravel” on Github, there are 281,126 repository results. It is easy to see that the

Laravel community is huge. That make software developing process become easier. There will be a lot of help and support.

b) Nodejs

Based on (Nodejs) and (Express) homepage

(1) Definition

Nodejs is an open-source, cross-platform runtime environment, it is designed for server-side application. It has built-in JavaScript programs that can run on OS X, Windows and Linux.

Expressjs is a framework built on top of Nodejs. It offers powerful features for web or mobile development. Expressjs supports HTTP and middleware methods which makes API extremely powerful and easy to use.

(2) Advantages and Disadvantages

Advantages:

- The remarkable feature of Node.js is that it receives and processes multiple connections with a single thread. This helps the system to consume the least RAM and run the fastest without creating new threads for each query like PHP. In addition, taking advantage of the non-blocking I / O advantage of Javascript that Node.js makes the most of the server's resources without creating latency like PHP.
- JSON APIs With event-driven, non-blocking I / O (Input / Output) and Javascript model are excellent choices for Web services made of JSON.
- If a programmer is going to write a Single page Application, then NodeJS is a good fit. With the ability to process multiple Request / s simultaneously fast response time. The applications written will not have to reload the page, including many requests from users that need fast performance to show professionalism, the NodeJS will be your choice.
- Shelling tools unix NodeJS will make full use of Unix to operate. That is, NodeJS can handle thousands of Processes and return 1 thread that makes the performance of the operation to the maximum and the best.

- Streaming Data Regular web pages send HTTP requests and receive responses (Data Stream). Assuming that you will need to handle an extremely large data stream, NodeJS will build Proxies to partition the data streams to ensure maximum operation for other data streams.
- Real-time Web applications With the advent of mobile applications & HTML 5, Node.js is very effective when building real-time applications such as chat applications, network services. Social like Facebook, Twitter, ...

Disadvantages:

- Nodejs is quite resource intensive. When doing a big task like video conversion, Nodejs is a bad choice because it can overload memory.
- Nodejs is not faster than PHP if it's all CRUD (Create Read Update Delete)
- Most of the Nodejs APIs will operate in the non-blocking / async method, so if you do not understand it clearly it will make the choice of Nodejs wrong.

(3) *Usage and popularity*

Not because it's hot and new, so Nodejs does everything well, for example, an application needs high stability, complex logic, PHP or Ruby languages ... is still a better choice. Here are the possible and should be written in NodeJS:

- Websocket servers: Web socket servers such as Online Chat, Game Server ...
- Fast File Upload Client: programs that upload files with high speed.
- Ad Server: The ad servers.
- Cloud Services: Cloud services.
- RESTful APIs: these are applications that are used for other applications through the API.
- Any Real-time Data Application: Any real-time speed requirement. Micro Services: The idea of micro services is to break a large application into small services and connect them together. Nodejs can do this well.

c) Choosing the suitable technology for Backend

Apparently the best option is Nodejs. Because Nodejs has a very fast development speed. With the express framework, backend development time when using Nodejs is greatly reduced by its ease of use. Also, the size of the server when using Nodejs is also much smaller when using Laravel

5. Speech Synthesis Markup Language (SSML)

a) Definition

According to (Taylor, A Markup Language For Text-To-Speech Synthesis, 1997)SSML stands for Speech Synthesis Markup Language. Simply put, it is like HTML but HTML helps define interface elements, SSML helps define the sound. SSML was created to create a common standard for regulating parameters.

SSML is a speech synthesis markup language developed by W3C, it is one of the standards applied in speech synthesis systems, and this language is built with XML foundation, which helps to analyze and process data more easily.

Since SSML is based on XML, it must follow the structure and syntax of XML, so the SSML namespace must be clearly defined, and <http://www.w3.org/2001/10/synthesis> is the standard SSML namespace and the following are some of the specific components of SSML.

- **Speak:** This attribute is a required member of SSML, it is used to specify an SSML file and define a namespace.
- **The emphasis tag:** This tag allows us to require emphasis on certain points in the text, with the emphasis tag giving us a level attribute for the user to require the emphasis.
- **Prosody tag:** This tag allows us to request controls for pitch f0, speed, and loudness of the voice, it includes several properties such as pitch, countour, range, rate, duration.

b) Reason of choosing SSML

To change the sound, normally, people will use audio modifier applications, apply effects to that audio. In addition, developer can also use code to edit audio, there are many libraries that support audio editing. However, if the above measures are applied, the development time will be very long. Taking advantage of the similarity between SSML and HTML, I will redefine SSML

tags so that the browser can read them as HTML while still following SSML standards. This makes the change of sound entail the change of the picture.

IV. Requirement analysis and Plan for RubySmile Text2Speech System

A. Review of other similar applications

1. Balabolka

Balabolka is a program that runs on the window form platform with the function of converting text to speech (TTS). All voices used in the Balabolka application are computer voices installed on your system. The application converts by allowing users to enter text directly on the application interface, which can then be heard directly or saved as an audio file. In addition, the application can read content from clipboard, read text from document file format, customize fonts and background color, read from system tray with hot keys. Balabolka supports a lot of text file formats such as: DjVu, DOC, DOCX, PDF, PPT, PPTX, EPUB, HTML, RTF, WPD, XLS, XLSX ...

The program uses Microsoft Speech API (SAPI) that allows users to change parameters of the voice, including speed and pitch... In addition, users can apply some special features. difference to improve pronunciation quality of voice, for example: change the spelling of words, correct pronunciation errors using syntax of common phrases, attach audio files to text ...

However, according to a review website (Justin, 2013) evaluating Balabolka software, there are a few weaknesses such as:

- Some custom voice functions still have errors, and the ability to customize is limited.
- Due to the installation on windows, there are times when the application is mistakenly recognized as a virus and users have a lot of trouble installing.
- Balabolka is limited by the language and voice pre-installed in the system, which greatly reduces the user's customizability.
- Simple UI makes it be boring and unattractive.

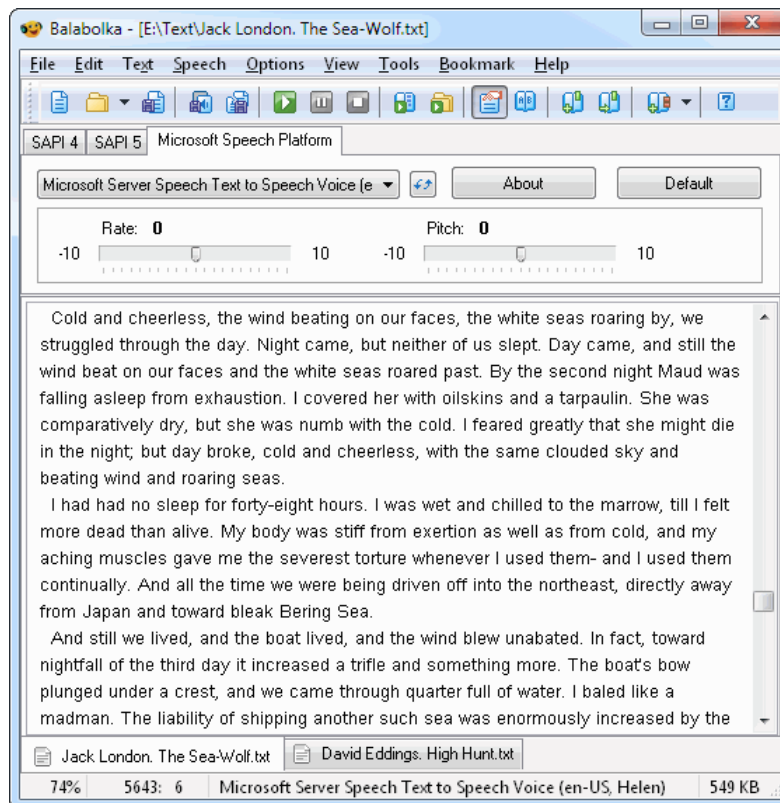


Figure 10: Interface of Balabolka Application

2. Panopreter Basic

Text-to-speech software allows you to get audio files to read all text on your computer. Panopreter is a text-to-speech software that runs on Windows forms. The application supports reading any text with many different ways, users can open any software window just copy to the Windows clipboard. Moreover, Panopreter can also convert text into various audio file formats such as mp3, wav, ogg and flac, which is very convenient for listening or using audio files for many different purposes...

Panopreter uses Microsoft's default voices that comes pre-installed on the Windows operating system, then through user editing, the application converts the selected text into an audio file. Audio files can be previewed or stored in a variety of formats.

With the convenience and ease of use of Panopreter, the app is aimed at businesses, educators, students, writers, and language learners.

Refer to the feedbacks on (UserFeedback, n.d.), users have rated Panopreter as follows:

- Panopreter makes installation difficult for users using Windows 10. Window defender mistakenly recognizes Panopreter as a virus. In some cases, errors occur during installation.
- Since Panopreter uses Window's built-in voice, Panopreter's voice is limited. In some cases, Panopreter is unable to retrieve voices from the system.
- Panopreter's interface is also quite simple, not really appealing to users.
- The quality of the output file is not stable, the output file is too heavy for the users' expectation, the quality does not meet the expectations.

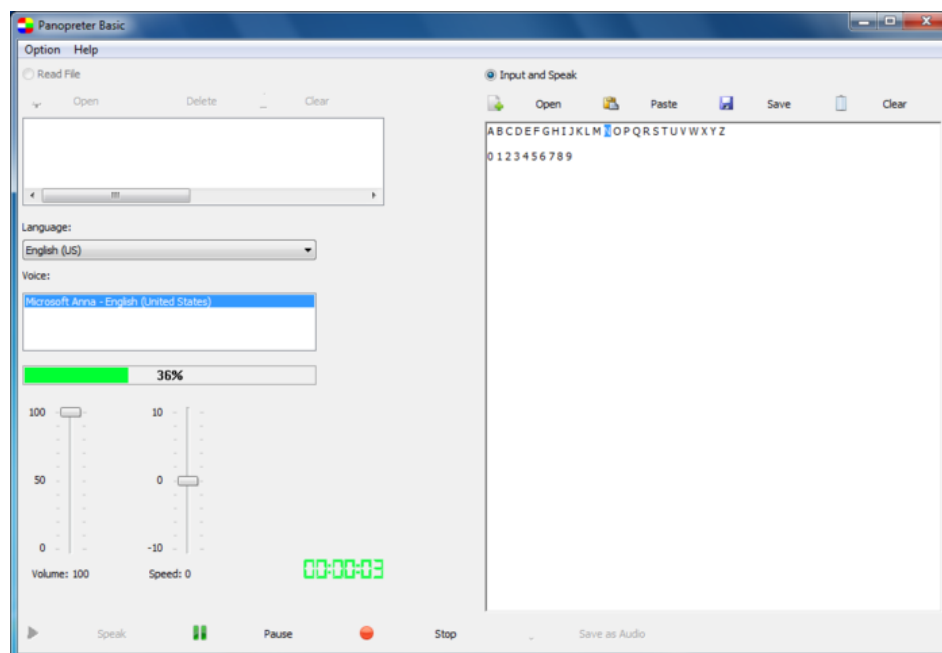


Figure 11: Interface of Panopreter Basic Application

3. Analysis of applications in the market

After analyzing 2 apps with similar purposes as RubySmile Text2Speech, both of these apps have the following in common:

- Both are window applications: this is inconvenient for the user. Many users do not want to download the software when they need to use it. They can also run into trouble during installation and use. The installation on Windows is dependent on the system. Browser-

based applications are often more convenient. Users can use anywhere, anytime with just an internet connection.

- Unfriendly interface: The interface of both applications is based on winform, this is a simple interface and is completely unattractive.
- Customization is not high: The user is limited to using the pre-installed voices on Windows. Whether the application runs well or not depends a lot on compatibility. In many cases, the application is blocked by the security system. In addition, users can only use 1 voice at a time, this causes quite a lot of trouble if the text appears in 2 or more languages.

B. Vocal Analysis

Speech characteristics are closely related to emotional state and speaker characteristics. The focus of this section is the theoretical analysis of human voice data. According to document (Hildebrand, 2020), all sound waves are described according to four properties: time, amplitude, frequency and spectrum. As shown below, the same word "Hello" but 2 ways to say it gives 2 different analysis results and 2 different emotions.

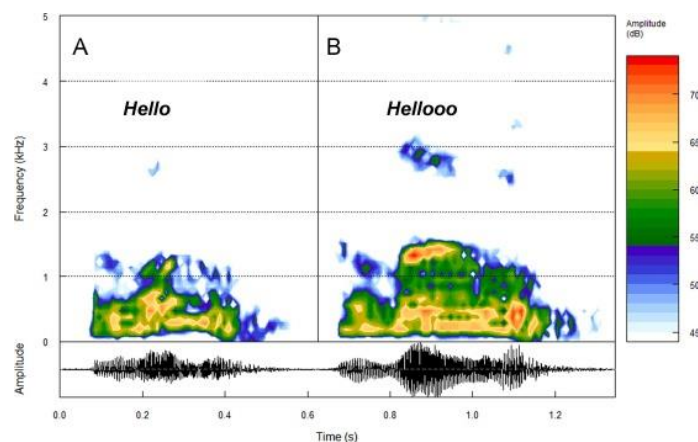


Figure 12: Hello and Hellooo sound wave (Hildebrand, 2020)

Imagine the pronunciation of the words "Hello" and "Hellooo". "Hello" is pronounced at normal speed, without emphasis, giving listeners a cold feeling. In contrast, "Hellooo" with extended "o", emphasized pronunciation in the first sound gives the listener the feeling of being excited by the speaker.

Citing 4-dimensional analysis of voice in document (Hildebrand, 2020)

SOUNDWAVE DOMAIN	PRIMARY VOCAL FEATURES (EXAMPLE METRIC)	LISTENER PERCEPTION	INFERRED STATES AND TRAITS BASED ON EXPRESSED SPEECH
TIME	Duration (Milli-/Seconds)	Duration of an Utterance	Anger†, Fear†, Sorrow†
	Speech and Articulation Rate (Words per Second)	Velocity of Speech	Anger†, Competence †, Contemplation†, Dominance†, Enthusiasm†, Extraversion†, Fear†, Happiness†, Persuasiveness†, Sadness†, Stress†, Tenderness†
	Voice breaks (Percentage Unvoiced Frames)	Number and Duration of Pauses	Competence†, Contemplation†, Extraversion†
AMPLITUDE	Intensity / Power (Sone)	Loudness of Speech	Aggression†, Anger†, Annoyance†, Dominance†, Extraversion†, Fear†, Happiness†, Tenderness†, Sadness†, Shyness†, Stress†
FREQUENCY	Variability of Intensity / Power (Sone Variance)	Loudness Variability	Anger†, Dominance†, Fear†, Happiness†, Sadness†, Tenderness†
	Fundamental Frequency (Hertz)	Pitch	Anger†, Competence†, Confidence†, Empathy†, Extraversion†, Fear†, Happiness†, Nervousness†, Persuasiveness†, Sadness†, Stress†, Tenderness†, Trustworthiness†
	Variability of Fundamental Frequency (Hertz Variance)	Pitch Variability	Anger†, Extraversion†, Happiness†, Sadness†, Shyness†, Sociability†, Tenderness†
SPECTRAL	Vocal Shimmer (cycle to cycle deviation from mean amplitude)	Loudness Perturbations	Anger†, Confidence†, Joy†, Stress†
	Vocal Jitter (mean absolute difference between consecutive μ s periods)	Pitch Perturbations	Anger†, Annoyance†, Happiness†, Sadness†, Stress†
	HNR (additive noise in signal in dB)	Voice Quality	Confidence†, Happiness†, Interest†, Lust†, Pleasure†
	Vocal Entropy (Shannon evenness of frequency spectrum)	Diversity of Vocal Transitions	Low mood†

Figure 13: A Four-Dimensional Conceptual Framework of Speech

Focusing on these 4 characteristics of vocals, Text2Speech application will be built with features to help users easily create voices according to their desires. The basic features would be:

- Break - Break between words.
- Speed - Change the speed of the segment.
- Emphasis - Change rhythm in a sentence.
- Loudness - Change the loudness of the voice.

C. Limitations of current Text-to-speech technology and SSML solution

1. Limitations of current Text-to-speech

In this section, to learn about the limitations in current text-to-speech technology, a universal application with text-to-speech functionality will be launched for study and testing. The most popular known application and widely used by many people currently is the Google translate application. Text reading functionality is included in this application. This is an easy-to-use function, just enter text and click the speaker icon, Google will immediately read the text entered.

After using Google Translate's text-reading function, there are a few easy-to-see problems as follows:

- Problem breaks in sentences: Google does not have the ability to make proper breaks in sentences. Although there is a break in dots and commas, but when reading the whole text, this break is not reasonable. The received voice does not feel genuine.
- Emotional problem: Similar to the problem this report is referring to, the reading voice is received quite poor emotionally in a sentence. This is the clearest sign between the mechanical voice and the human voice.
- Wrong reading: In case of wrong reading, the word has a special reading method or the word is an acronym. Google can only read common words and appear in the dictionary. As for words that are commonly used in communication, Google cannot recognize them and can easily become misread.
- Can not read multiple languages, multiple voices: This is the problem that was mentioned in the idea of building this application. Current text-to-speech applications with automatic translation do not support inserting multiple languages or voices in a text. For the conversion of stories to audio files, the conversation in the story has only one voice, reducing the attractiveness of the story, the user becomes more difficult to follow the story content when listening to the audio.

D. User requirement definition

To clarify the user needs of this application, I created a survey to collect customer requirement. This collection will help me get a better idea in app design.

Narrow down the scope to increase the effectiveness of the survey, here are 1 areas of survey that were taken:

- Target: people between 16 and 30 years old
- Method: Do an online survey
- Area: On the social network facebook, at facebook groups about making youtube, making movies, reading books, ...

The survey questions include:

- What is your current job?
- Have you ever had a need for voice recording at work or in life?
- How much is your need to use these voices?
- When you had to record your voice, what difficulty did you face?
- Do you know some tools that use text to speech technology?
- Have you ever tried using tools to get voices for your product?
- Are the tools that you have been using for fees?
- What are the problems you face using these tools?
- What is the first thing of a good application you will notice?
- If it's free, would you be willing to change from voice recording to using text-to-speech technology?
- There is going to be a text-to-speech application. Do you contribute anything to the application?

After doing a survey within 2 days, the survey had 85 responses. Here is a list of the answers in multiple choice format:

1) What is your current job?

This question was created for scaling the customer size. What job can the customers be? The survey is send to the media group, content creator group, youtubers, but there are many people are student, freelancer and movie maker.

2) Have you ever had a need for voice recording at work or in life?

92.6% of survey participants have a need to record voice. With this relatively large ratio, it proves that the need to use voice in everyone's life and work is very much. Building an application in this area will attract a lot of users.

3) How much is your need to use these voices?

This question is to refer to the user's usage needs. With 55.6% of people with medium level and only 11.1% of people with heavy usage, it is easy to see that the demand of users is at a medium level.

4) When you had to record your voice, what difficulty did you face?

There are a ton of different answers, but the most submitted ones are:

- It takes a long time to do the recording work.
- They had problems with their recording equipment such as settings, and the recording was contaminated with noise, which resulted in a lot of processing time.
- The recorded voice was not what they wanted.

5) Do you know some tools that use text to speech technology?

96,3% people know about text-to-speech tools mean that there are many people know about this technology. The market size is huge and there are many people can be customer of this application

6) Have you ever tried using tools to get voices for your product?

In 96% people know about text-to-speech tools, there are 85,2% people have used this text-to-speech technology for their purpose. 11,1% have learnt about it. The

user's need is huge and there is much challenge. How different can RubySmile Text2Speech make to impress the users?

7) Are the tools that you have been using for fees?

29.6% answered that they do not pay any fee for using these tools. 37% of them have to pay but only when they use a lot. And 33.3% is a complete fee. In application development maintenance, in order for the application to be kept up-to-date and maintained, there is a fee required to maintain this activity. RubySmile Text2Speech application is not out of that. Applications will be developed with free functions to attract users and will have advanced functions to obtain application maintenance costs.

8) What are the problems you face when using these tools?

There are many issues mentioned by the user and these issues are spread out with a lot of answers. The typical answers are:

- They don't know how to use it and don't understand how to use it. These tools are very difficult to use for them.
- They have problems with their voices. There are dialects they just can't have. They cannot change the voice in a piece of text. The audio they receive cannot be edited directly on the converting application, but have to take the audio editing software to continue another step, which is very time consuming.
- The tone in the audio they received was very bad, it was not real and made them really feel disappointed. There is absolutely no editing function they want.
- When using the paid functions, although they are advanced functions, the efficiency is not high for them, which makes them not want to use text-to-speech applications anymore.
- Received audio is easily recognizable not being read by humans.

9) What is the first thing of a good application you will notice?

This question is to survey users' interest in using an app. 73.1% of users think that a good application is an easy application to use. 23.1% of users think an application is good or not depends on whether the application's interface is friendly or not. The ease

of use is a big issue. It will take time for researching, especially for a new application in the market such as the RubySmile Text2Speech application.

10) If it's free, would you be willing to change from voice recording to using text-to-speech technology?

Everyone loves a free app. RubySmile Text2Speech will be a free application, but for advanced functions there will be an additional fee.

11) Soon, there will be a new text-to-speech application. Do you contribute anything to the application?

A review of history of successful applications will need to be built on community input. Typically, the game Half Life was built according to the needs of the community, the opinions of the community are included in the game, which has made the game successful. Even though the game was created in 1998, it is still hot today. Facebook too, starting with public opinion that created a big wave.

The collected comments are as follows:

- The application interface should be beautifully designed, easy to use and stable
- Make voice diversification, customize your voice easily and choose from a variety of tones
- Because the tone of the voice can be heard at times. In some cases, a slight adjustment of the pitch of the voice creates completely different meaning for a sentence. They will be happy when the new system schematics or displays voice output pitch parameters. This will allow them to know in more detail when to increase / decrease the pitch of the voice.
- The application can give suggestions for voice editing

E. Summary and Future Plan

After survey and research, it is easy to see that the user demand for a text-to-speech application is quite large.

From surveys to similar applications and user surveys, to sum up, the RubySmile Text2Speech application needs the following features and features to be able to attract more users:

- Application needs to be able to edit user voice tones. The basic elements that need to be included in the editing function will be: Break, speed, emphasis, volume.
- The application needs to have other editing functions that currently do not have, such as: Changing the way of reading a word, changing the language and voice in a document.
- Easy to use for users. Users using the application need to understand what they need to do and what to do. This comes from the UI / UX design for the app. The app is easy to understand and easy to use, the more it attracts users.

However, there are many challenges to face. The biggest challenge is: How to build a user-friendly application with the easiest operation to use? With a development team consisting of only 1 person, currently, building an easy operation experience is very difficult, it is also very time consuming. The only solution right now is to let the users do it. This release version will be an experimental version, the application will come with feedback function to get user feedback. From those opinions build a complete application over time.

There is short-term and long-term goal

	Period	Goals
Short-term	4 months	Experimental version with basic text-to-speech function Feedback feature
Long-term	2 years	Completed version with full feature

For feature feedback, Google sheet API will be a good choice. There are 2 characteristics that make Google sheet selected:

- The Google sheet API is very easy to use

- Google sheet is fully functional with excel statistics and the application does not need a database to store for this test version

V. Design of RubySmile Text2Speech System

A. Architecture Diagram

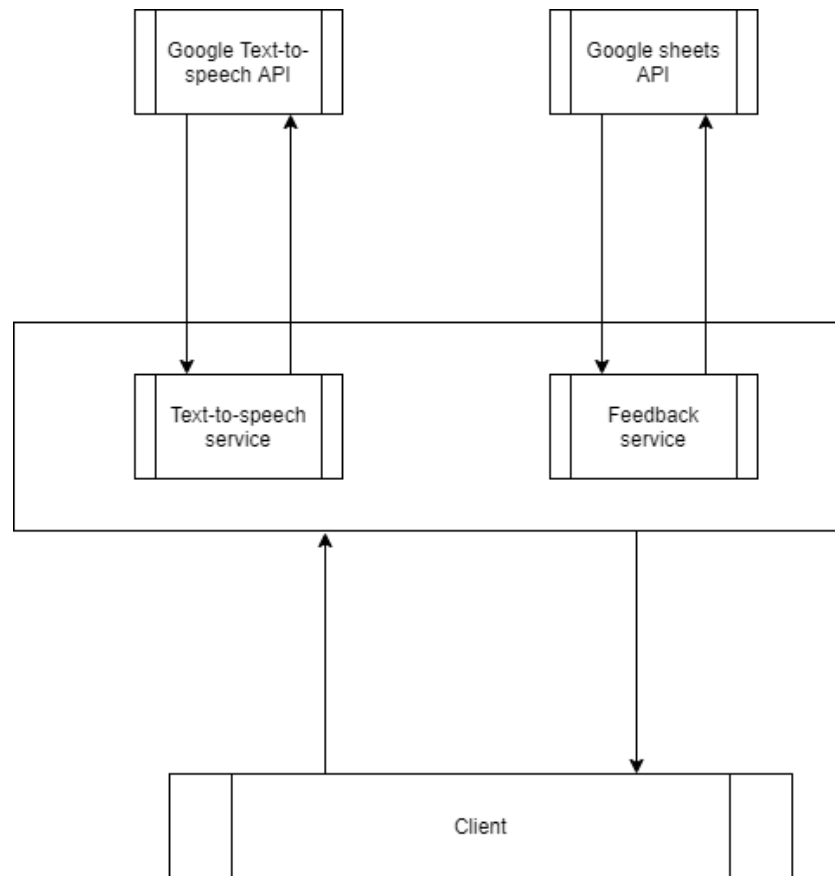
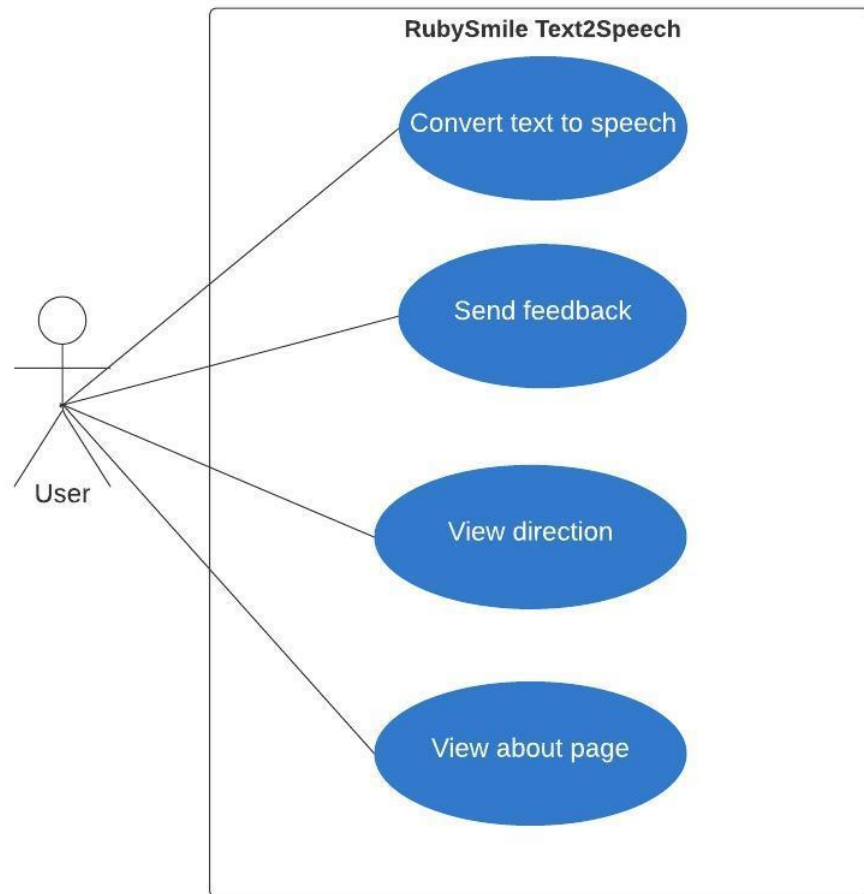


Figure 14: Architecture diagram of RubySmile

RubySmile Text2Speech application is built on web architecture. According to the above diagram, users will directly interact with the application interface, including the Text-to-speech service and the Feedback service.

These 2 services will send a request to Google API to receive the desired response. Google's 2 APIs are Google Text-to-speech and Google sheet

B. Usecase



The application basically works quite simply, users have 4 main functions that can be used:

- Convert text to speech with customizations made by users.
- Send feedback for RubySmile Text2Speech.
- Display 2 pages of HowToUse and AboutUs

C. Sequence Diagram

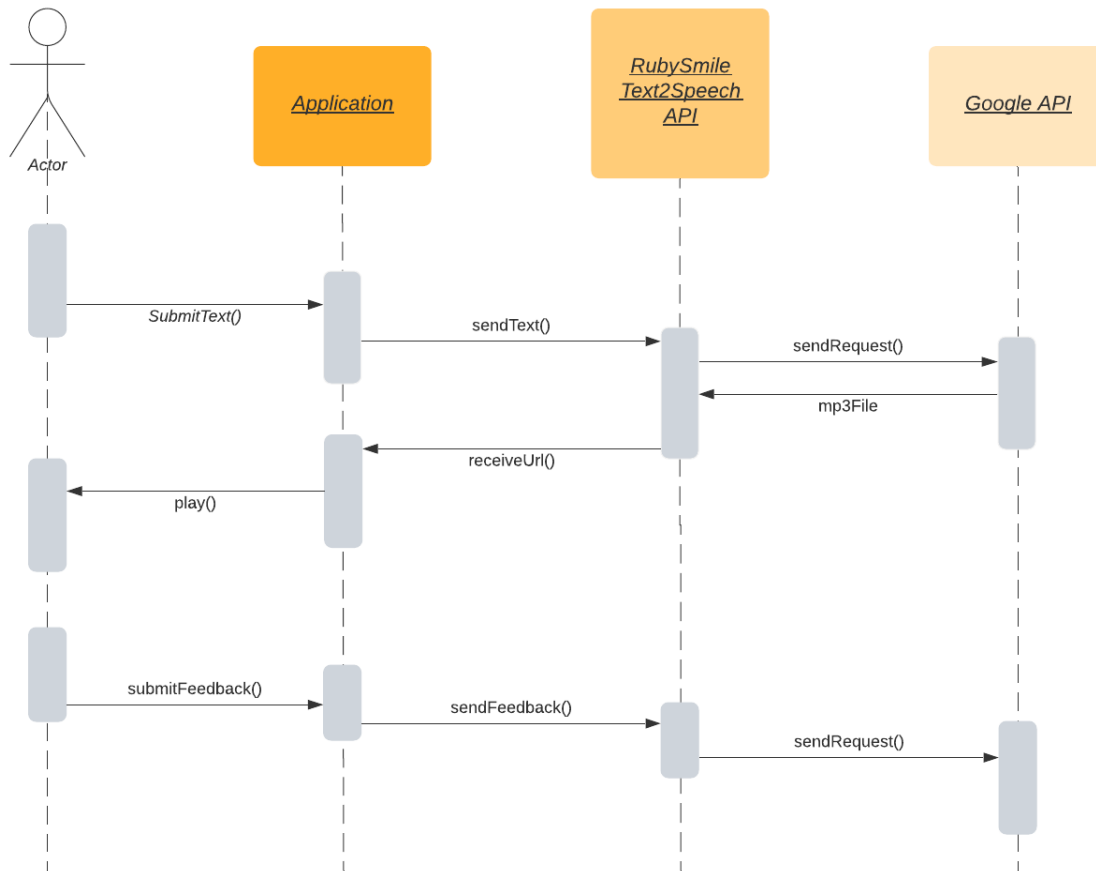


Figure 15: sequence diagram

According to (Poranen, 2003), A sequence diagram is a diagram that shows all information related to functions such as the interaction between objects through the function and the results of those interactions, the order in which those functions are performed. The sequence diagram contains information such as input, output, loop and results, etc. In this system, for the customer, application, API and Google API. Customers are the ones who use this application to convert text to speech, they enter text into the application. This app will then send the request with that text to the application API. The application API continues to send requests to the google API. Google API returns to the application API and the application API sends back to the application, users simply press play to listen to the audio. Similar to the feedback submitting function.

D. Wireframe

In the test version of the RubySmile Text2Speech app, the app will have only four main screens: Text-to-speech, Feedback, HowToUse and AboutUs.

- Text-to-speech display is a screen with text-to-speech function and allows users to edit voice with the customization on it.
- Feedback screen is a screen that allows users to send feedback about the application.
- The HowToUse screen is a tutorial screen that shows the user how to use the application.
- The AboutUs screen is the application introduction screen.

1. Text-to-speech page

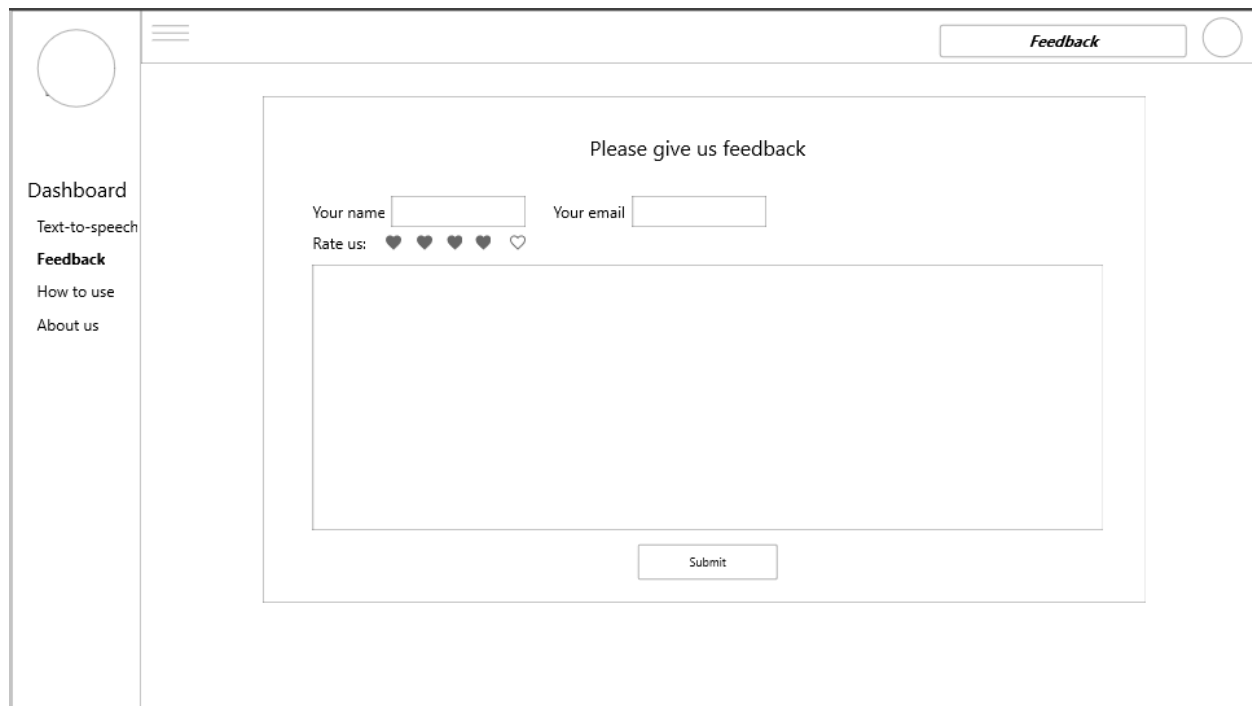
Figure 16: Wireframe Text-to-speech page

Based on the requirements analyzed in the above sections, this page will have the full range of features as follows:

- Search text: this area will allow users to search for the text they are looking for.
- Audio play area: After finishing editing and submitting text, the app will allow users to listen to the audio by pressing play or clicking the download button.

- Text input area: Here the user will select the voice and language and then enter the text to be converted into the application. To use the voice editing, users need to highlight the paragraph they need to edit. To add another language or voice, the user presses "Add voice". Click submit to initiate the conversion.
- Options area: After selecting the text, the user will click on the effect option that the user wants in this area.

2. Feedback page



The wireframe shows a web page layout for a feedback form. On the left is a sidebar with a circular profile icon at the top, followed by a hamburger menu icon. Below these are links: "Dashboard", "Text-to-speech", "Feedback" (which is bolded), "How to use", and "About us". The main content area has a title bar at the top with a "Feedback" button and a circular icon. The main heading is "Please give us feedback". Below this are two input fields: "Your name" and "Your email". Under the "Your name" field is a "Rate us:" label followed by five heart icons, the first four of which are filled. Below these is a large rectangular text area for the feedback comment. At the bottom center of the form is a "Submit" button.

Figure 17:Wireframe Feedback page

The function of this page is quite simple. Users just need to enter their name, email, rating from 1 to 5 and feedback or comment in the text box, then click submit.

3. HowToUse and AboutUs page

The function of these 2 pages is almost nothing. Users simply click to see the information on the pages and read them.

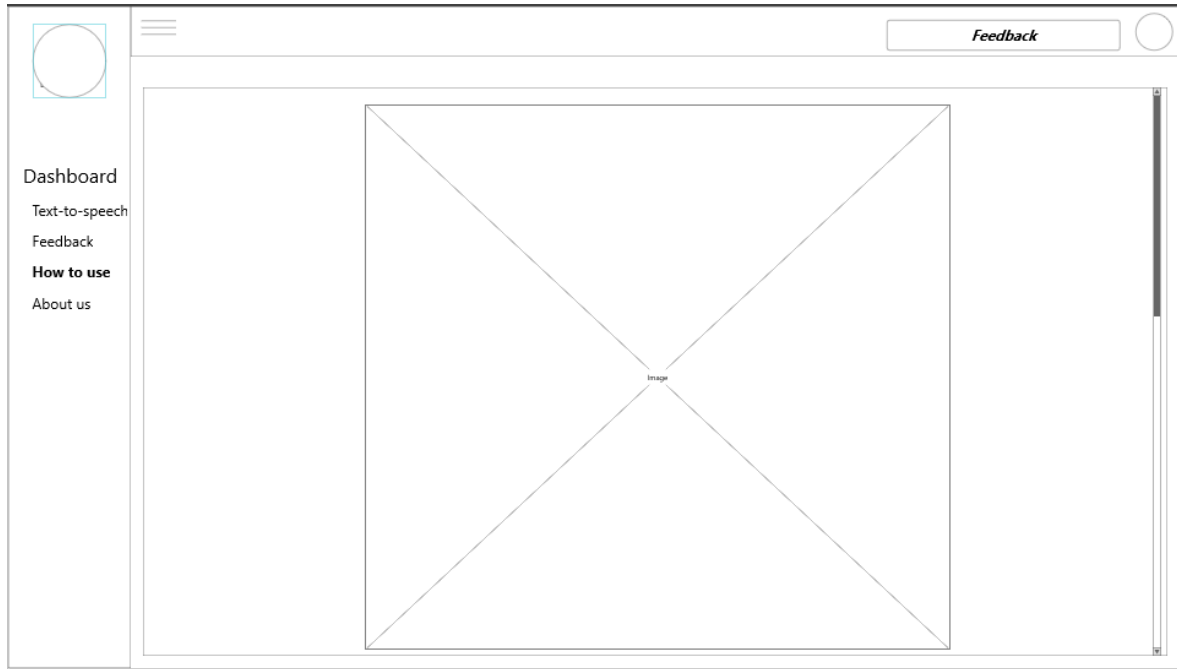


Figure 18: Wireframe "How to use" page

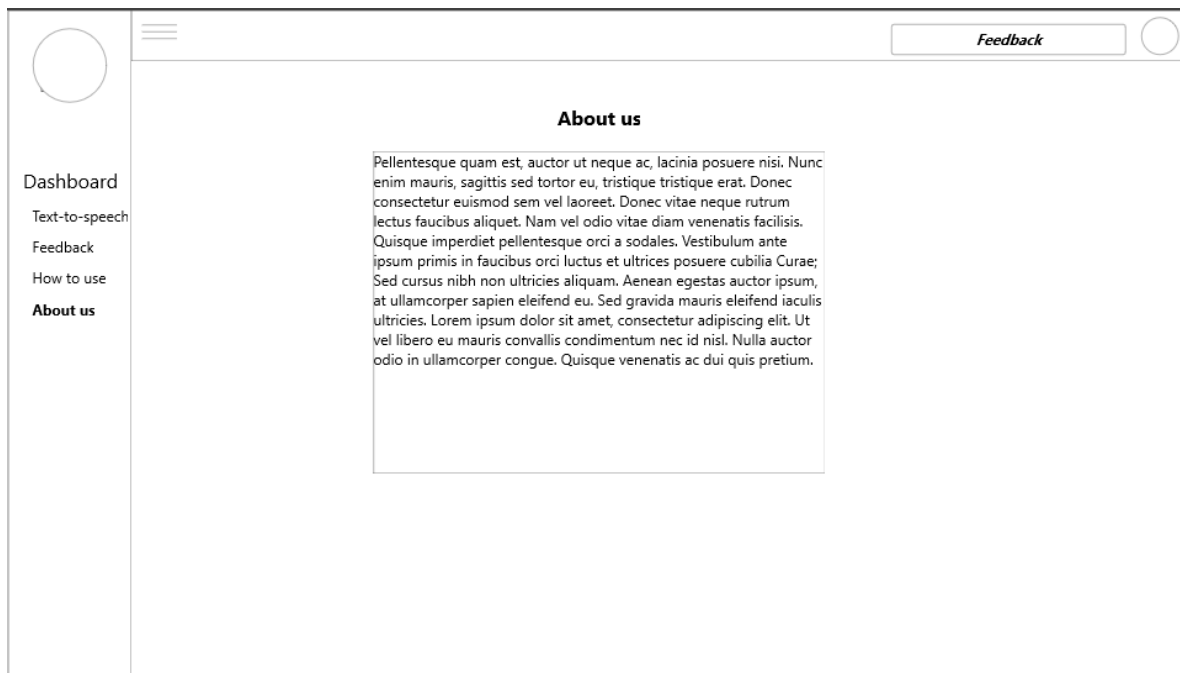


Figure 19: Wireframe "About us" page

VI. Development of RubySmile Text2Speech System

A. Frontend

1. The idea of manipulation on the interface

Inspired by highlighting text in Microsoft Word, this test version will allow users to highlight text to edit desired properties in voice.

For example: Hello, I am Ha Ngoc Linh. I am a student but I have already built an application.

Users may highlight the text like this:

Hello, I am Ha Ngoc Linh. I am a student but I have already built an application.

In the sentence, red highlight mean emphasizing, blue highlight means changing speed. Black space means breaking

2. Structure

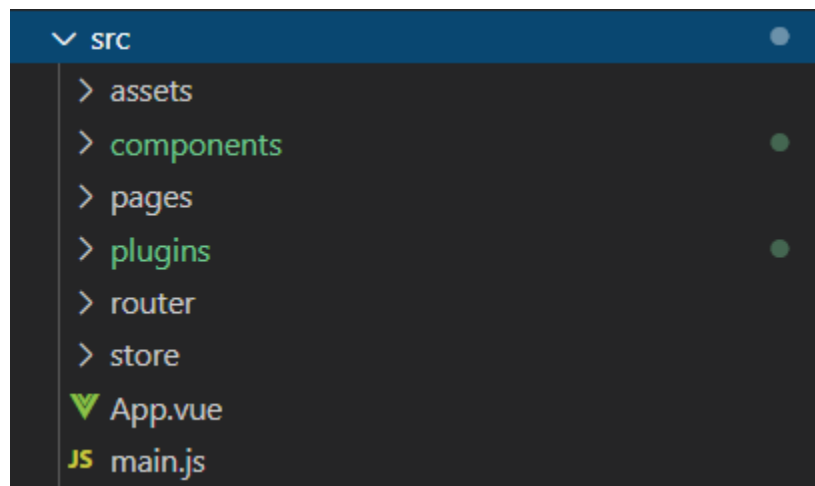


Figure 20: Source code structure

The entire source code will be in the src directory, so in the whole section, only the src directory will be parsed. The following is the src directory structure.

In the src folder includes folders: assets, components, helper, pages, router, store and 2 files App.vue and main.js.

- App.vue: This is the entry point component. This is the place where all other components will be initialized. Literally it is the main file of the project.
- main.js: Entry point file to mount App.vue. This is the file that renders the App.vue component.
- assets: This is where you will work with Webpack, which is also the place to store project's images, icons, style css files, ...
- components: This is the directory that contains the UI components of the project
- router: This is where it will write the routes and connect them to Components.
- Store: This folder contains Vuex store files. Vuex store is used for storing state of SPA application.
- plugins: This directory contains javascript files. This is where to inject functions to Vue files or contents for project.
- pages: This directory contains all view pages of application.

a) pages

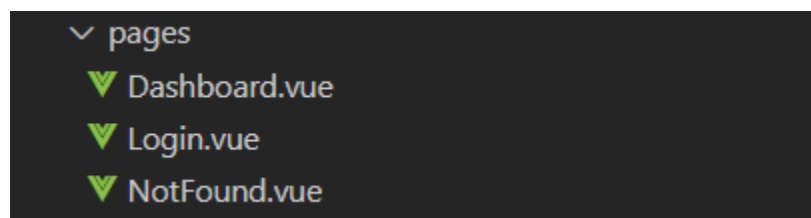


Figure 21: source code of pages folder

In this version, the design style used will be the Admin template, so the pages will include left side-bar and top nav-bar. Particularly for the NotFound page, this page will be displayed when a user accesses a route that does not exist in the application, this page will not include the side-bar and the nav-bar. Page Dashboard will be the main page, the interfaces inside the Dashboard will be placed in the components folder. (Login page in this folder is a test code)

b) components

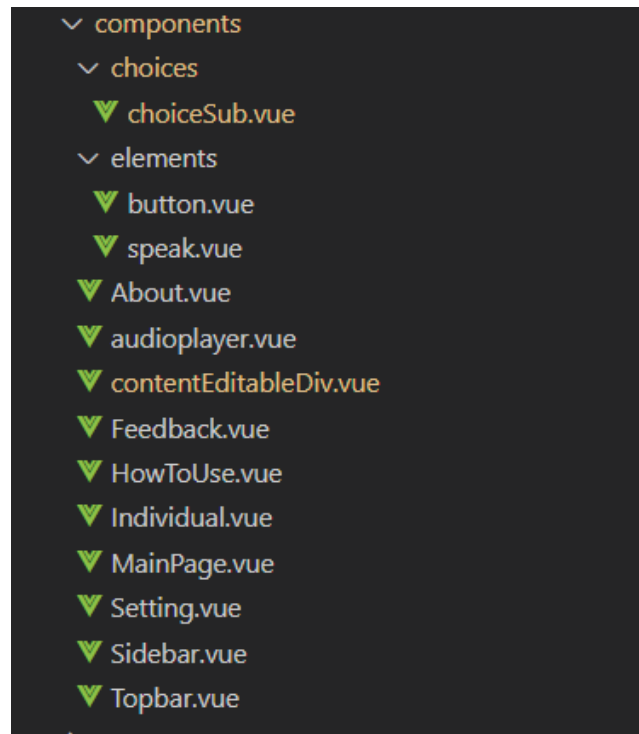


Figure 22: source code of components folder

This is the directory where the main interface of the application is stored (The files that are not mentioned are the ones that are currently under development and not yet in use).

- For the AboutUs page: File About.vue
- For the How to use page: File HowToUse.vue
- For the Feedback page: File Feedback.vue
- For the page Text-to-speech: MainPage.vue is the main interface file of text-to-speech page. audioplayer.vue is a component of this page, this component is an audio player. contentEditableDiv.vue is a component used to enter text. choices/choiceSub.vue is the component file that shows the interface of options choosing effect for text.
- Topbar.vue and Sidebar.vue files are 2 common components of all pages, it is the file that displays the side-bar and top nav-bar of the application.

c) Assets

In this directory there are all styling files for the application, including CSS, SCSS, fonts, icons and images files.

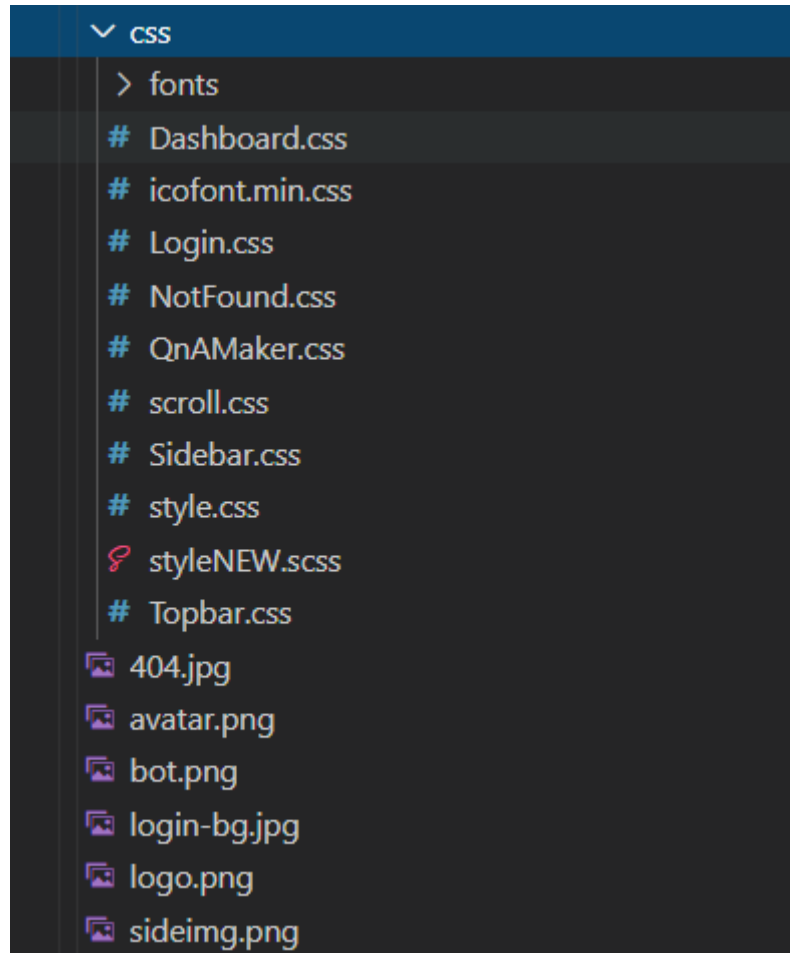


Figure 23: source code of asset folder

d) Router

Here, the application routes will be defined, because of this directory, the application can run as a SPA (Single Page Application).



Figure 24: source code of router folder

e) Plugins

This directory contains js files. api.js and axios.js define the API caller. The vuejs files just call the function in the api.js file and can easily call the API. As for the ./lib folder, this defines the SSML structure and the functions for SSML.

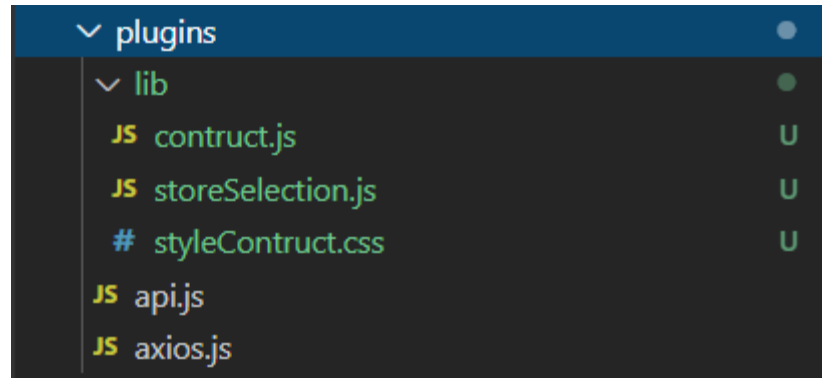


Figure 25: source code of plugins folder

f) Store

The store directory is the directory of Vuex. Here will contain state management functions.



Figure 26: source code of store folder

3. Code flow

a) *lib/construct.js*

This is the top important file of this project. It is the constructor of selected SSML text which is the text with special value. The application sends SSML text to text-to-speech API then the API will return an audio file which has a special effect at the selected text.

```
const constructorSSML = {
  id : 0,
  p : {
    isDouble : 1,
    isActive : 0,
    tag : "p",
    attributes : {},
    class: 'pp',
  },
  paragraph : {
    isDouble : 1,
    isActive : 0,
    tag : "paragraph",
    attributes : {},
    class : "paragraph",
  },
  say_as : {
    isDouble : 1,
    isActive : 0,
    tag : "say-as",
    class : "say_as",
    attributes : {
      interpret_as : {
        isActive : 0,
        text : "interpret-as",
        value: "",
        options: ["date","time","literal","telephone","currency","cardinal","ordinal","digits"]
      }
    }
  }
}
```

```
    },
    format : {
      isActive : 0,
      text : "format",
      value: "",
      options: []
    },
  },
},
emphasis : {
  isDouble : 1,
  isActive : 0,
  tag : "emphasis",
  class : "emphasis",
  attributes : {
    level: {
      isActive : 0,
      text: "level",
      value: "",
      options: ["strong", "moderate", "none", "reduced"]
    }
  }
},
prosody : {
  isDouble : 1,
  isActive : 0,
  tag : "prosody",
  class : "prosody",
  attributes : {
    pitch : {
      isActive : 0,
      text: "pitch",
      value: "",
      options: ["x-high", "high", "medium", "low", "x-low", "default"]
    }
  },
}
```

```
    contour : {
      isActive : 0,
      text: "contour",
      value: "",
      options: []
    },
    ranger : {
      isActive : 0,
      text: "ranger",
      value: "",
      options: ["x-high", "high", "medium", "low", "x-low", "default"]
    },
    rate : {
      isActive : 0,
      text: "rate",
      value: "",
      options: ["x-fast", "fast", "medium", "slow", "x-slow", "default"]
    },
    duration : {
      isActive : 0,
      text: "duration",
      value: "",
      options: []
    },
    volume : {
      isActive : 0,
      text: "rate",
      value: "",
      options: ["slient", "x-soft", "soft", "medium", "loud", "x-loud", "default"]
    }
  },
  break : {
    isDouble : 0,
    innerText : "_",
```



```
class : "break",
isActive : 0,
tag : "break",
attributes : {
  strength : {
    isActive : 0,
    text: "strength",
    value: "",
    options: ["none","x-small","small","medium","large","x-large"]
  },
  time : {
    isActive : 0,
    text: "time",
    value: "",
    options: []
  }
},
audio : {
  isDouble : 0,
  isActive : 0,
  innerText: "audio",
  class : "audio",
  tag : "audio",
  attributes : {
    src : {
      isActive : 0,
      text: "src",
      value: "",
      options: []
    }
  }
},
desc : {
  isDouble : 1,
```

```

    isActive : 0,
    tag : "desc",
    attributes : {}
  },
  mark : {
    isDouble : 1,
    isActive : 0,
    tag : "mark",
    attributes : {
      name : {
        isActive : 0,
        text: "name",
        value: "",
        options: []
      }
    }
  }
}
}

module.exports = constructorSSML;

```

This module defines which SSML is activated and which attribute value of the SSML is chosen. There are also all options of a specify SSML has. Additionally, this module defines the styles for the SSML tag. The text in the SSML tag will get the corresponding style defined by the style class of that SSML tag.

```

graph TD
    subgraph Backend
        B[Backend Nodejs API]
    end
    subgraph Frontend
        subgraph TTS_Feature [Text-to-speech feature Frontend]
            subgraph Plugins_API [plugins/api.js  
plugin/axios.js  
API calling function]
            end
            subgraph MP_Vue [MainPage.vue  
Text-to-speech Page]
            end
            subgraph CTD_Vue [contenteditableDiv  
(MainPage.vue)  
text editor]
            end
            subgraph ChoiceSub_Vue [choices/choiceSub.vue  
onClickSave()  
Save selected text with SSML to state  
onClickRemove()  
Remove SSML from the selected text]
            end
            subgraph ChoiceSub [choiceSub]
            end
            subgraph Lib_Construct [lib/construct.js  
SSML constructor]
            end
            subgraph Plugins_StoreSelection [plugins/storeSelection.js  
saveSelection()  
save selected text into state  
restoreSelection(range)  
restore selected text which is stored in state]
            end
        end

        B -- "send request" --> Plugins_API
        Plugins_API -- "receive response" --> B
        MP_Vue -- "submit text with SSML" --> Plugins_API
        Plugins_API -- "audio url" --> MP_Vue
        MP_Vue -- "User Input text" --> CTD_Vue
        CTD_Vue -- "Text" --> ChoiceSub_Vue
        CTD_Vue -- "Change style text" --> ChoiceSub_Vue
        ChoiceSub_Vue -- "select text" --> ChoiceSub
        ChoiceSub -- "select SSML for selected text" --> Lib_Construct
        Lib_Construct --> ChoiceSub_Vue
        MP_Vue -- "play audio" --> A_Vue[aplay  
vue-aplay  
play audio file]
        A_Vue --> MP_Vue
    end

```

The above Diagram explains the code of the text-to-speech feature.

After entering the text, the user will perform the operation to highlight the text to select the text you want to change the voice characteristics. There are 2 buttons with 2 events `saveSelection()` and `restoreSelection()`. `saveSelection()` stores that selection in the state. `restoreSelection()` will retrieve the text stored in the state and select that text in `contenteditableDiv` (highlight the text).

After selecting the text that needs to be edited, the user performs the selection of speech properties for that text. The choiceSub component (defined in choiceSub.vue) will display the SSML tag choices, those taken in lib / constructor.js. After selecting the SSML tag, the system will give you a choice of attributes for that SSML tag. The user who clicks save will run the onClickSave() function, which will attach the SSML tag with the selected attribute to the text in contenteditableDiv, at the same time information about the text and the SSML is also saved in the state. If a user clicks on a text tagged with SSML, information about that text will be removed from the state and displayed to the user. If you want to make changes, the user will have to click remove to run the onClickRemove() function, which will remove the tag information in the state and remove the SSML tag from the text.

Choosing an SSML tag for that text will also change the style of that text in contenteditableDiv because that text is in a tag that is already class style.

This process is repeated until the user has finished editing.

After editing is complete, the user clicks submit to run the onClickSubmit() function, which will fetch all of the content including the text and the SSML tag in the contenteditableDiv to send the request to the server. The request sending functions are built into two files plugins / axios.js and plugins / api. axios.js to define API methods such as GET, PUT, POST, DELETE. The api.js file defines the functions that call api.

After calling the api at the Nodejs server, the server will return an audio url, which is the result of the text-to-speech conversion. This audio url will be assigned to the aplay component (this is a component defined to play audio files). Users can listen to it by clicking the play button. The user can also download that file.

c) Feedback

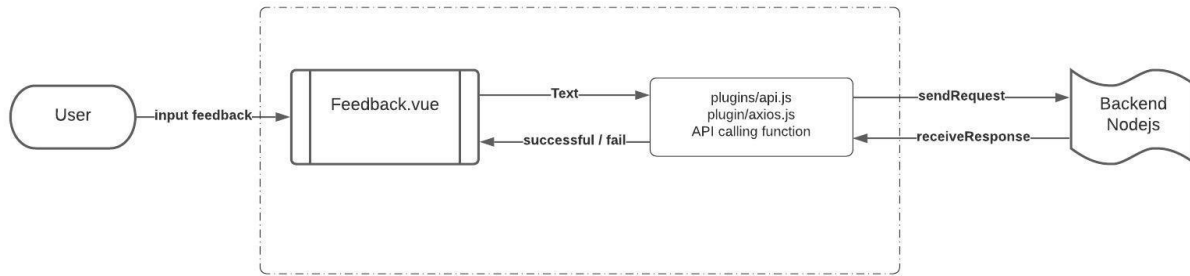


Figure 28: Code Flow feedback feature

The mechanism of this function is quite simple. Users access the Feedback page and enter feedback. After clicking submit, the application will call the request sending function in api.js and send the request to the API backend. If the feedback is successfully saved, the backend will return the response to a successful status. The user will receive a notification according to the submit status of the application.

d) HowToUse and AboutUs

For the reason that these 2 pages are only used to display non-changeable information. Therefore, these are 2 static pages and absolutely no function connecting to the backend.

B. Backend

1. Structure

For this version, security feature for API will be not available.

API server was built with MVC design pattern, by using this pattern, the code can be clean and easy to maintained

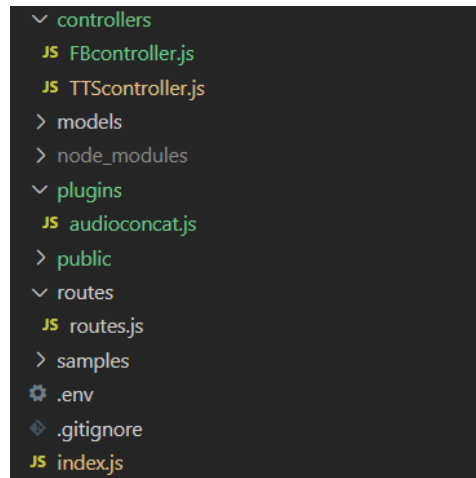


Figure 29: structure code of backend

2. Code flow

The index.js file will be the main file of the backend. In this file will define the libraries and launch it together with the project. Project is run under config in .env file (Normally it will run under port 3000).

routes / routes.js will receive the request sent to the corresponding API address and then call the functions contained in the controllers.

There are 2 controllers that are FBcontroller.js (Feedback) and TTScontroller.js

FBcontroller, after receiving a request from the frontend, it will continue to send the request to the Google Sheet API and wait for the response status. If successful, it will return the response to the frontend is successful.

TTScontroller, after receiving a request including SSML text from frontend will run a loop (SSML text is cut into different segments by voice and language) send SSML text to Google text-to-

speech API. In the loop, each time it receives an audio file, the application will store that file in folders samples with unique name. End of the loop (complete conversion of the entire SSML array), the application will perform the concatenation of audio files together using plugins / audioconcat.js and delete the newly created files in the samples folder, the new file will be saved. in folder “public”. “public” folder is a special folder, users can access all files in this folder. When the whole process is complete, the audio file link will be sent back to the frontend.

VII. Testing

A. Test plan

No	Description	Date	Tester
1	Text-to-speech feature	10/11/2020	Ha Ngoc Linh
2	Feedback feature	12/11/2020	Ha Ngoc Linh
3	HowToUse and AboutUs page	14/11/2020	Ha Ngoc Linh

B. Test case

No	Test case	Test title	Details	Expected result	Actual result	Status	Note
1	Feedback submit	Invalid input	Leave an empty value in form	Unable to submit	Unable to submit	Pass	
2		Invalid input	Wrong format email	Unable to submit	Unable to submit	Pass	
3		Valid input	Normal submit	Successful	Successful	Pass	
4		Failure submit	Submit with error (create an error in backend)	Warning failure submit	Failure warning	Pass	
5	Text to speech	Single SSML	Selected text with single SSML	Successfully convert text to speech with chosen SSML effect	Successfully convert text to speech with chosen SSML effect	Pass	

6		2 different SSML – 1 text	Selected text with 2 different SSML	Successfully convert text to speech with chosen SSML effect	Successfully convert text to speech with chosen SSML effect	Pass	
7		2 same SSML – 1 text	Selected text with 2 same SSML but different option	Successfully convert text to speech with chosen SSML effect	Cant convert	Fail	
8		2 different SSML – 2 text	Selected 2 text with 1 SSML each	Successfully convert text to speech with chosen SSML effect	Successfully convert text to speech with chosen SSML effect	Pass	
9		Text SSML with special characters	Text with special characters (\$, %, ^,...) with SSML	Successfully convert text to speech with chosen SSML effect	Successfully convert text to speech with chosen SSML effect	Pass	
10		SSML with 2 different voices	Create text and choose 2 different voice with 1 language	Successfully convert text to speech with chosen SSML effect	Cant convert	Fail	This function havent been done
11		SSML with 2 different language	Create text and choose 2 different language	Successfully convert text to speech with chosen SSML effect	Cant convert	Fail	This function havent been done
12							

C. Summary

The application only passed 8 out of 11 tests. The level of perfection after performing the test is quite low. Approximately 40% of the proposed functions are not working. The reason for this situation is the lack of time and inaccurate planning, which makes the project not complete as expected.

The basic functions are to convert text to speech at a basic level only. This means that RubySmile application will only function in its correct state when each text is only associated with a unique SSML. With 2 or more SSMLs, there will be an error.

The feedback function works quite correctly, with all tests of this feedback feature working well.

There is also one more reason that the project was changed midway. Initially in the period of 1 year of project development, the project theme was completely different from the present one. The project was changed to RubySmile Text2Speech when development time was cut in half. This greatly affects product quality and progress.

VIII. Evaluation

A. Human Interaction

Based on (Mirkowicz, 2018) There are 10 usability heuristics for user interface design

1. Visibility of system status

The design should always inform the user of what the application is doing, when the user knows the current system state, they know what to do next. In the RubySmile Text2Speech application, the pending states (with text loading) are displayed on the screen when the application is loading the page or loading the text to speech conversion result, ... This helps the user know what are they doing, did they do correctly and what should be done.

2. Match between system and the real world

Design should use familiar words, phrases, and concepts instead of internal jargon. Compliance with real-world conventions will cause information to appear in a natural and logical. SSML instead of being displayed as a tag is being displayed as color, these colors are associated with the intensity of the sound. However, this solution is being researched and surveyed for users to upgrade and improve gradually.

3. User control and freedom

Users often experience the situation of performing actions by mistake. They need a "cancel" button to get rid of the unwanted action without going through a lengthy process. With every submit button, the user after clicking will have a popup to confirm the action. In addition, when editing the SSML, the user will have enough add-edit-delete action for the text he just selected.

4. Consistency and standards

Actions in the RubySmile Text2Speech application are all consistent and similar. This increases the familiarity in the process of using with the user. All operations in the application have a common and consistent logic.

5. Error prevention

Good error messages are important, but careful best designs prevent the problem from happening in the first place. All input fields are validated. However, the validation of the text to

speech function is quite limited due to the limited level of not being able to validate the data before sending it, which makes errors often occur.

6. Recognition rather than recall

Humans have very limited short-term memory. An interface that forces the user to memorize, will reduce the amount of cognitive effort required from the user. The functions in the RubySmile Text2Speech application are all on a single interface page. When the user has edited SSML of a text, the user will be able to review the SSML of that text just by clicking. However, this will be limited by showing only the most recently added SSML. If the text is added with 2 SSML effects, the application will only display 1.

7. Flexibility and efficiency of use

Shortcuts - hidden for novice users - can speed up interaction for the experienced user and can cater to even inexperienced users. This allows the user to tailor frequent actions. However, this shortcut function does not work correctly on RubySmile Text2Speech application.

8. Aesthetic and minimalist design

Make sure that the visual elements of the interface support the primary goals of the user. All functions are designed according to its features. Because the feature is separated by pages, the visual elements of this feature are not related to other features.

9. Help users recognize, diagnose, and recover from errors

Error messages should be in plain language (no error codes), pinpoint the problem exactly, and suggest a solution in a positive way. This functionality is also quite limited in the text to speech converter. Since identifying bugs in this feature is quite complex, development of this functionality has not been possible.

10. Help and documentation

Help content and documentation should be easy to find and focus on the user's task. RubySmile Text2Speech application is developed exclusively 1 page with a manual feature (Page "HowToUse"). Since application is a new experience, this is a necessity of application.

B. Product review

Because the product has never been released, the RubySmile application has not received user reviews. However, the main purpose of this app version is to collect product reviews from users. From these reviews will give a completely different version, closer to the needs of the user.

For the individual developers, the application has not achieved the desired level of perfection. Due to unforeseen logic, the product is unfinished and not completed on time. In terms of function, the product achieved 60-70% of the plan. The main problem being encountered in the application is the user's manipulation of the text to speech function. The problems with this text to speech feature have all been addressed in the tests section of this report.

C. Plan analysis

The project plan was not carried out as planned. The fact that the project was changed the subject from education to text to speech when the old project was under development 40%. For this reason, the development time for the text to speech project has been cut by half. Initially as planned, the RubySmile project was calculated to only complete the code within 3-4 months, but due to a logic problem, the application was again interrupted and the plan had to be changed. 1 again in the development sequence. The finished condition of the product only reaches the level of 60-70%. This is a failure of changing the plan, changing the project topic.

IX. Conclusion

The process of creating RubySmile Text2Speech system includes many processes: research, design and development. Although the project was not as expected, it has left a lot of lessons

A. Analysis skill

According to the literature review, the process of doing research has brought about great success in knowledge of different technologies and software, in architecture and even in the scientific field of the connection between voice and emotion. contact. In addition, the research implementation process also brings the ability to analyze and choose technologies and solutions.

B. Development skill

While doing this project, I learned a lot about Vuejs, Nodejs, Google API, etc. With help and guidance from mentor - teacher Do Tien Thanh. I have improved my project analysis skills, programming skills, problem solving skills. In addition, I also learned the ability to meticulously plan a project, anticipate possible problems.

C. Report writing skill

The most learned thing is the skill of writing reports. With the mentor's enthusiastic guidance, I have learned how to structure a report, learned how to quote the source accurately, how to present a report professionally, ... This is very important in future projects.

X. References

- Amazon_Poly. (n.d.). *Amazon Poly*. From Amazon: <https://aws.amazon.com/blogs/machine-learning/tag/amazon-polly/>
- Angular. (n.d.). *Angular Doc*. From Angular: <https://angular.io/docs>
- Balabolka. (n.d.). *Balabolka* . From Balabolka : <http://www.cross-plus-a.com/balabolka.htm>
- Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology*, 5.
- Bauer, L. (2019). *Agile Methodology and System Analysis*. From University of Missouri–St. Louis: <http://www.umsl.edu/~sauterv/analysis/Agile%20Methodology%20and%20System%20Analysis.htm>
- BuiltWith. (n.d.). *Laravel Usage Statistics*. From Built With: <https://trends.builtwith.com/framework/Laravel>
- CaptiVoice. (n.d.). *CaptiVoice*. From CaptiVoice: <https://www.captivoice.com/capti-site//>
- Cavalcanti. (2020). *Angular vs React vs Vue: Which one will be popular in 2020*. From Morioh: <https://morioh.com/p/6decb109ae79>
- Express. (n.d.). *Express*. From <https://expressjs.com/>
- Geekboots. (2020, Jan 9). *Geekboots*. From Node JS vs Laravel: <https://www.geekboots.com/story/node-js-vs-laravel>
- Hildebrand, C. (2020). Voice analytics in business research: Conceptual foundations, acoustic feature extraction, and applications. *Journal of Business Research*, 364-374.
- Hughey, D. (2009). *Comparing Traditional Systems Analysis and Design with Agile Methodologies*. From University of Missouri–St. Louis: <http://www.umsl.edu/~hugheyd/is6840/waterfall.html>

Isard, A. (1995). *SSML: A Markup Language for Speech Synthesis*. University of Edinburgh.

Justin. (2013, February 11). *Turn text to speech with Balabolka, a portable text-to-speech reader*.

From dotTech: <https://dottech.org/96760/windows-review-balabolka-portable/>

Kappagantula, S. (2019, Nov 26). *Microservices Tutorial – Learn all about Microservices with Example*. From Edureka: <https://www.edureka.co/blog/microservices-tutorial-with-example>

Koch, A. S. (2004). *Agile Software Development: Evaluating The Methods For Your Organization*. Artech House Publishers.

Linguatec. (n.d.). *Linguatec Text-to-speech*. From Linguatec: <https://www.linguatec.de/en/text-to-speech/>

Mirkowicz, M. (2018). Jakob Nielsen's Heuristics in Selected Elements of Interface Design of Selected Blogs. *Social Communication*, 30.

Mishra, Minati. (2007). *Ethical, Legal and Social aspects of Information and Communication Technology*.

MobileScout. (2016, October 3). *Google Text to Speech can now play speech audio louder than background music - v3.10 released*. From MobileScout: <https://www.mobilescout.com/android/news/n74195/google-text-to-speech-update.html>

NaturalReader. (n.d.). *NaturalReader*. From NaturalReader: <https://www.naturalreaders.com/>

Newman, S. (2020). *Monolith to Microservices Evolutionary Patterns to Transform Your Monolith*. 1005 Gravenstein Highway North, Sebastopol, CA 95472: O'Reilly Media, Inc.

Nicholas Fearn, B. T. (2020, September 02). *Best text to speech software of 2020: Free, paid and online voice recognition apps*. From Techradar: <https://www.techradar.com/best/best-text-to-speech-software>

Nodejs. (n.d.). *Nodejs documentation*. From Nodejs: <https://nodejs.org/en/docs/>

Otwell, T. (n.d.). *Laravel Document*. From Laravel: <https://laravel.com/docs/>

Patel, H. (2019, January 20). *Essential Clients' Guide to Agile Development Methodology*. From News for public: <https://www.newsforpublic.com/agile-development-methodology/>

Pekka Abrahamsson, O. S. (2002). *Agile software development: Review and analysis*. Otamedia Oy, Espoo : JULKAISIJA – UTGIVARE.

Poranen, T. (2003). *How to Draw a Sequence Diagram*. Tampere University.

ReportsAndData. (2020, April). *Reports and Data*. From Audio Book Market: <https://www.reportsanddata.com/report-detail/audio-book-market>

RICHARDSON, C. (2019). *Microservices Patterns with examples in Java*. Shelter Island: Manning Publications Co.

Sumon, S. (2020, April 8). *Angular vs. React vs. Vue Detailed Comparison Guide - Which One to Choose in 2020*. From ThemExpert: <https://www.themexpert.com/blog/angular-vs-react-vs-vue>

Taylor, P. (1997). A Markup Language For Text-To-Speech Synthesis. *University of Edinburgh*, 20.

Taylor, P. (2009). *Text-to-speech synthesis*. New York: Cambridge University Press.

UserFeedback. (n.d.). From Freewarefiles: https://www.freewarefiles.com/review_9_105_48651.html

Voice_Dream_Reader. (n.d.). *Voice Dream Reader*. From Voice Dream Reader: <https://www.voicedream.com/>

XI. Appendix A – Source code

Project Github Link: [harnetlinh/FinalProject_HaNgocLinh \(github.com\)](https://github.com/harnetlinh/FinalProject_HaNgocLinh)

(https://github.com/harnetlinh/FinalProject_HaNgocLinh.git)

XII. Survey result

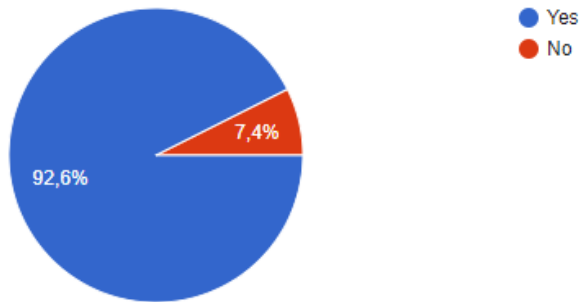


Figure 30: Have you ever had a need for voice recording at work or in life?

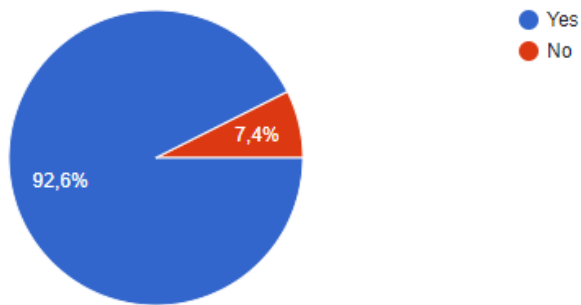


Figure 31: Have you ever had a need for voice recording at work or in life?

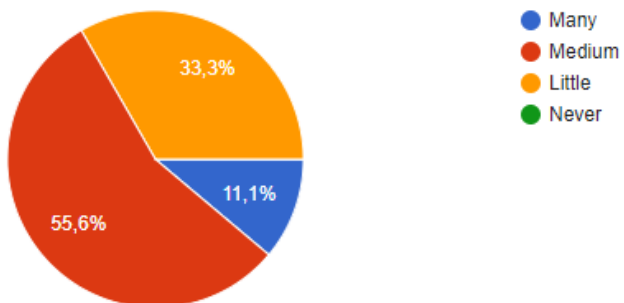


Figure 32: How much is your need to use these voices?

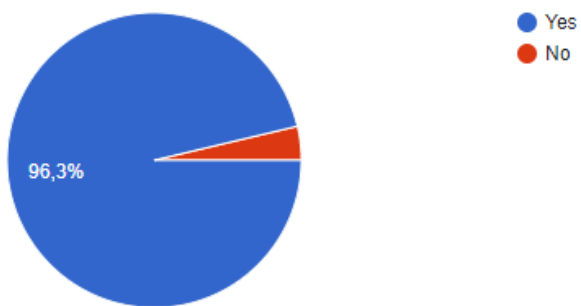


Figure 33: Do you know some tools that use text to speech technology?

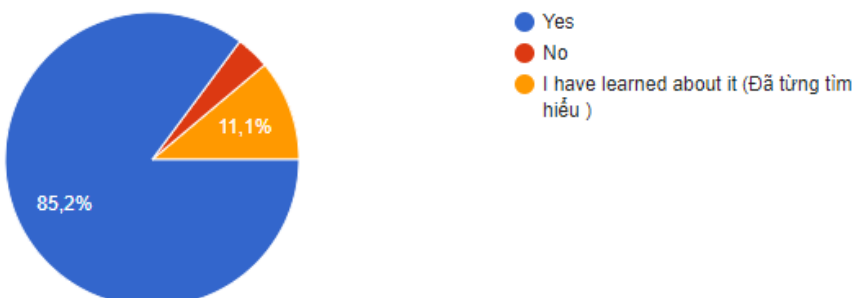


Figure 34: Have you ever tried using tools to get voices for your product?

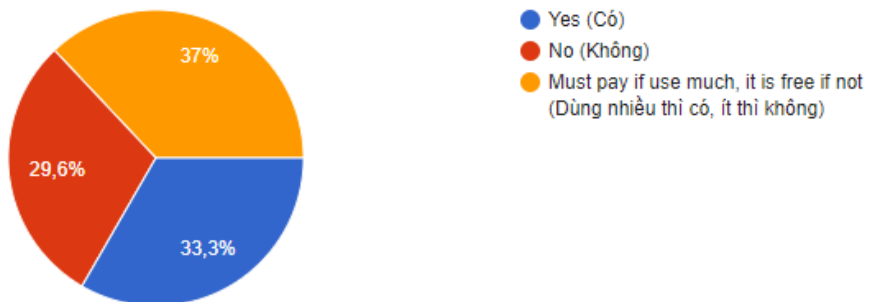


Figure 35: Are the tools that you have been using for fees?

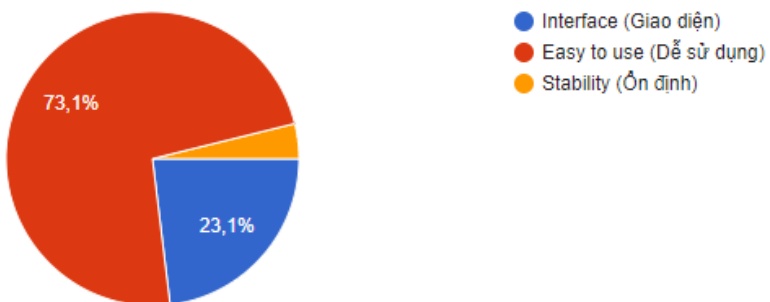


Figure 36: What is the first thing of a good application you will notice?

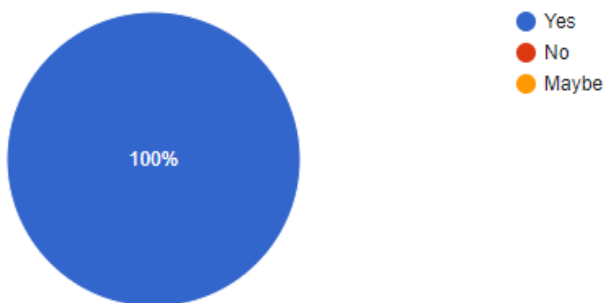


Figure 37: If it's free, would you be willing to change from voice recording to using text-to-speech technology?

XIII. Appendix C – Project Schedule

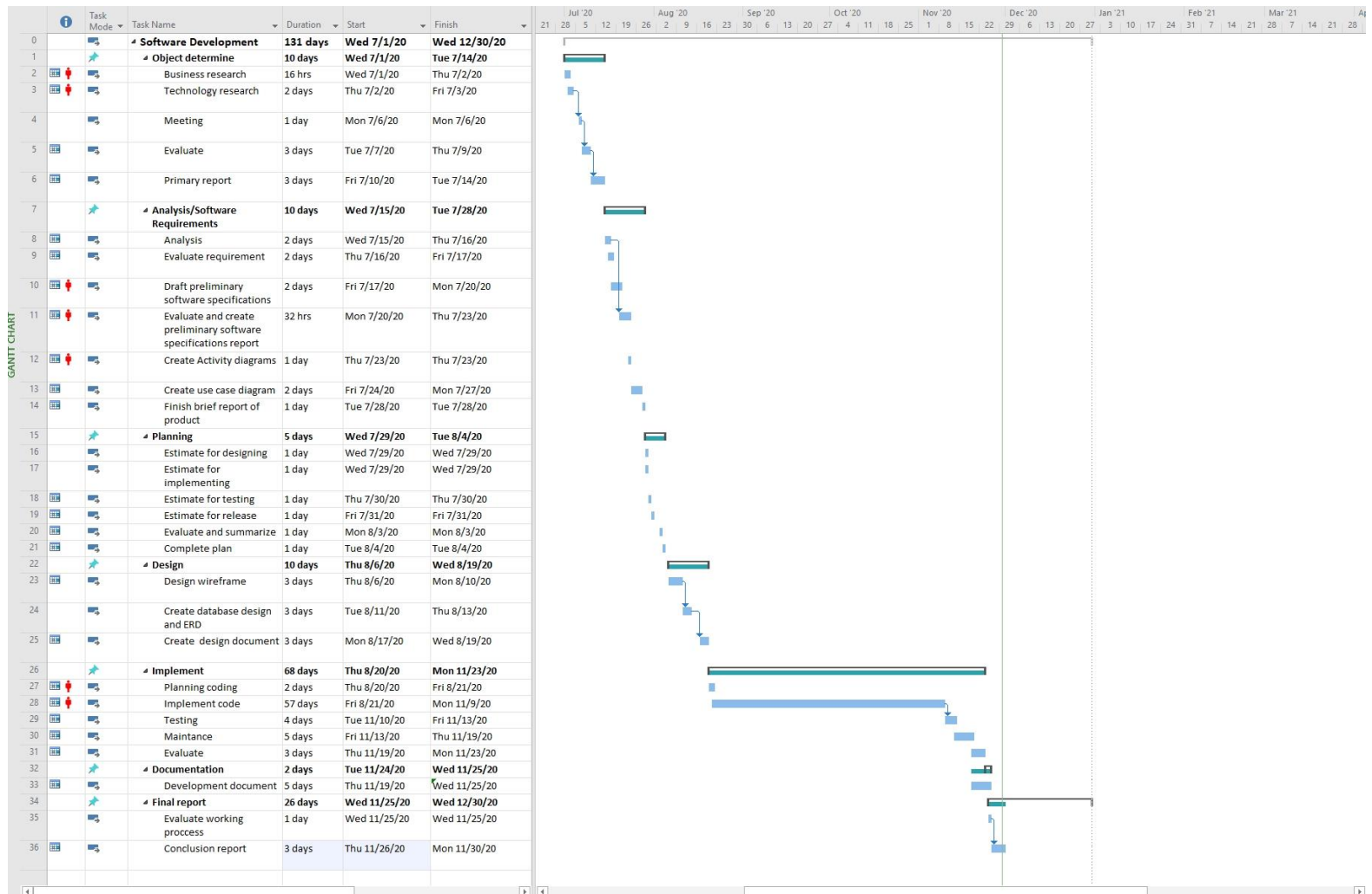


Figure 38: Project plan

The project plan RubySmile Text2Speech will only be changed in July. The reason for the change is that there are some shortcomings of the old project:

- The old project idea is so common, too many people did.
- Old project ideas are hard to do with something new.
- The old project idea could not develop much of the technology one wanted to learn.

When converting ideas to a RubySmile project, the author considered a lot before deciding and rescheduling. The RubySmile Text2Speech project idea will help the author challenge himself

more. RubySmile is also a new idea. After consulting with teacher Do Tien Thanh, receiving his support and help, the author decided to switch to the RubySmile Text2Speech project.

Although the project was not successful as expected and planned, the project helped the author learn a lot of skills, technology and lessons.

The plan was constructed on a waterfall model, which basically consisted of 6 phases

- Analysis
- Planning
- Design
- Code
- Testing
- Maintenance

XIV. Appendix D – Screen Captures

A. Text to speech screen

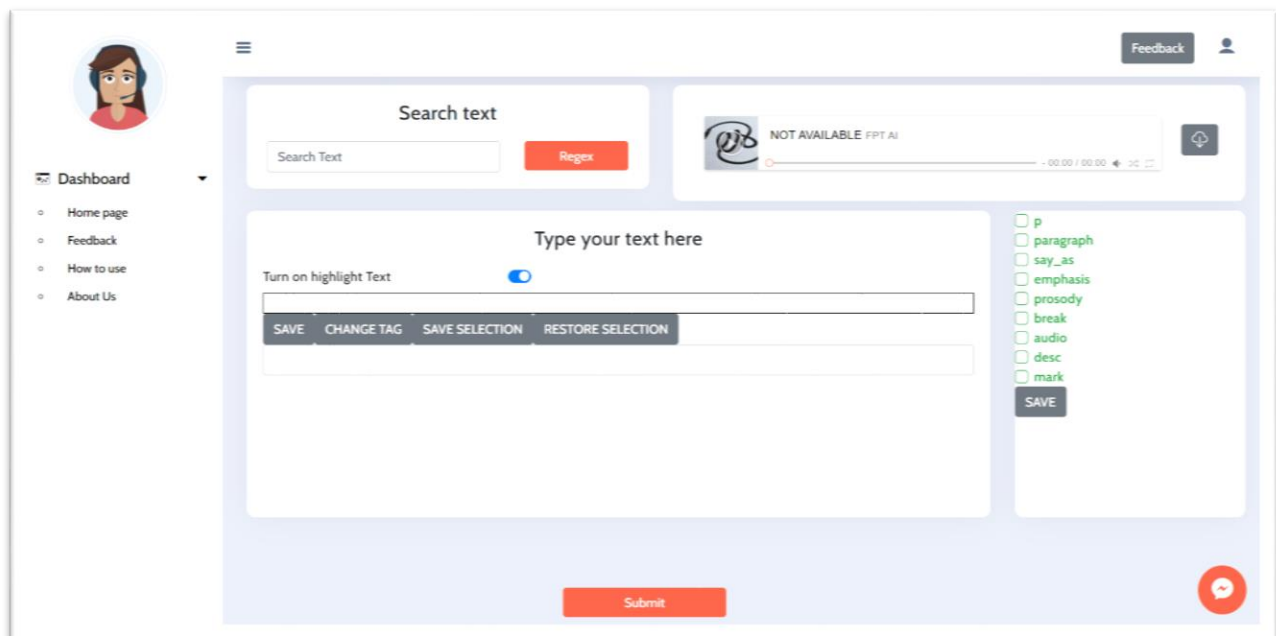
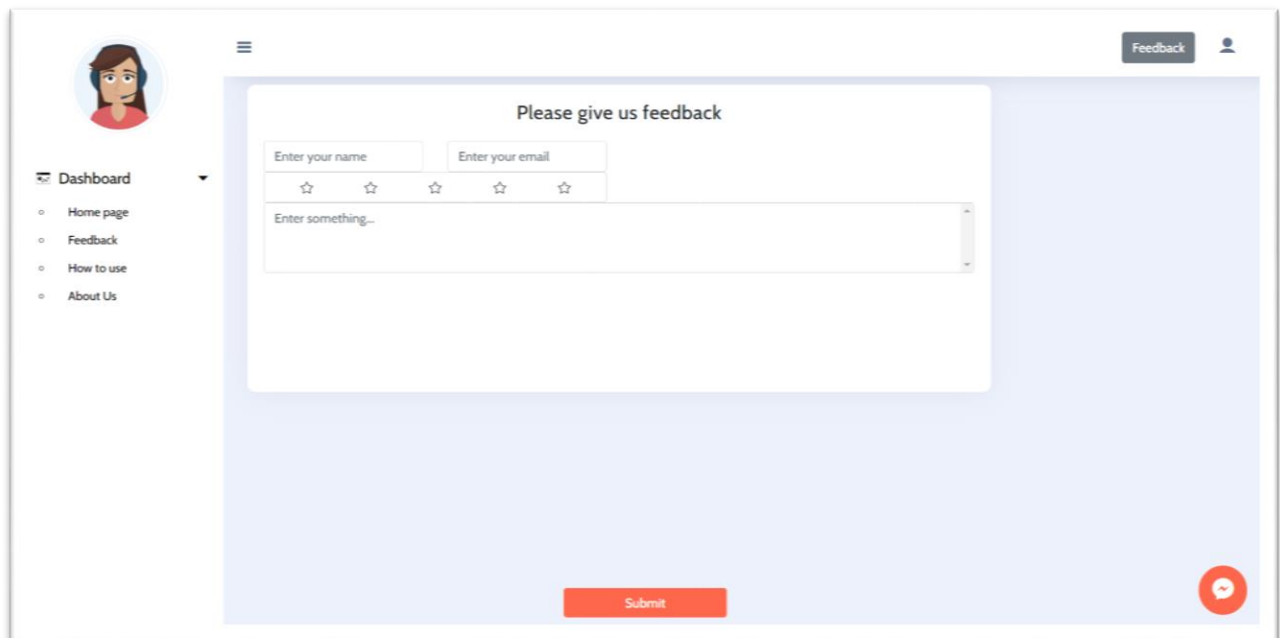


Figure 39: Homepage screen

B. Feedback screen



The Feedback screen features a light blue background. On the left is a sidebar with a user profile icon and a 'Dashboard' menu containing 'Home page', 'Feedback', 'How to use', and 'About Us'. The main content area is titled 'Please give us feedback' and contains a form with two input fields for 'Enter your name' and 'Enter your email', a row of five star icons, and a large text area for 'Enter something...'. A red 'Submit' button is at the bottom center, and a red speech bubble icon is in the bottom right corner. A 'Feedback' button and a user icon are in the top right.

Figure 40: Feedback screen

C. HowToUse screen

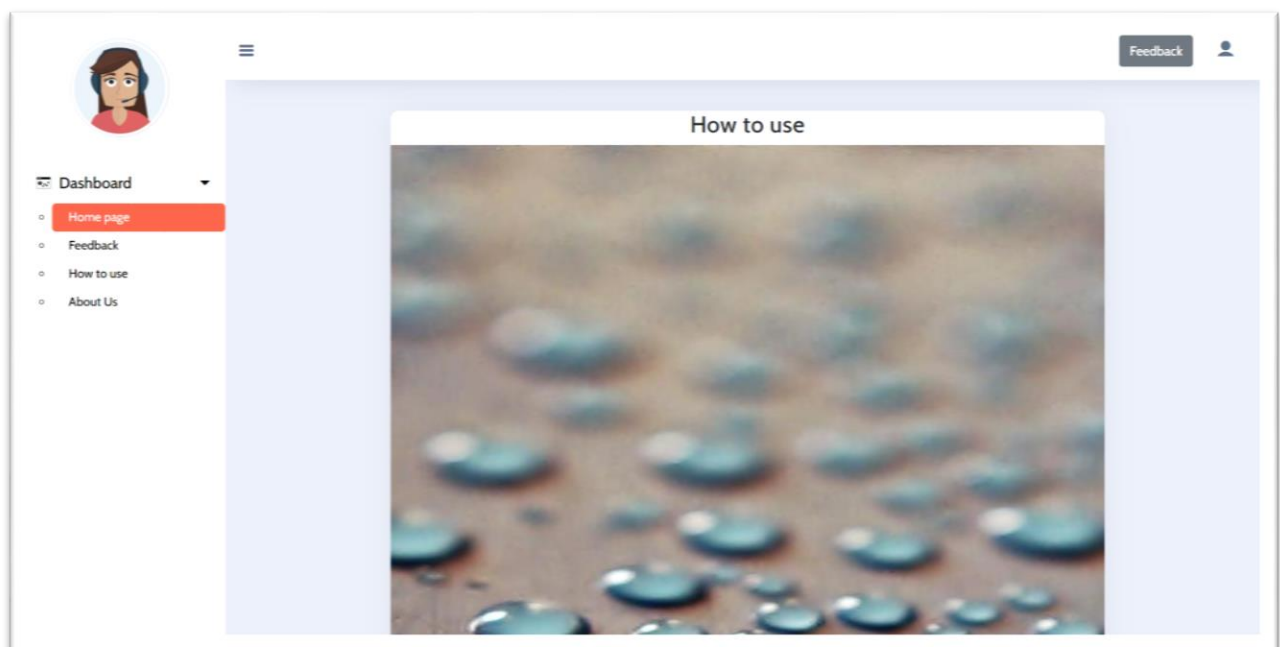


Figure 41: HowToUse Screen

D. About us screen

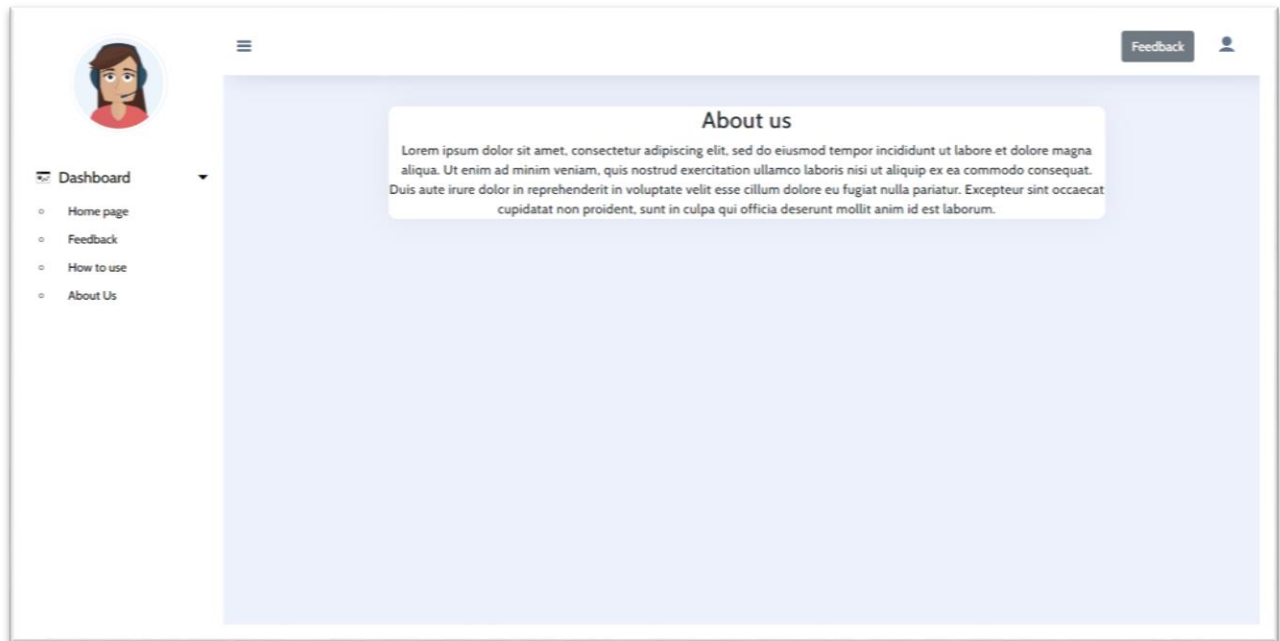


Figure 42: About us screen