HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

———————— * ————————

THESIS

SUBMITTED FOR PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

ENGINEER

IN

INFORMATION TECHNOLOGY

A WEB-BASED TOOL FOR CONSTRUCTING TEXT SUMMARIZATION CORPUS

Author: **Nguyen Anh Tu**
Class ICT-59

Supervisor: **Dr. Nguyen Thi Thu Trang**

HANOI, May 2019

HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

*

THESIS

SUBMITTED FOR PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

ENGINEER

IN

INFORMATION TECHNOLOGY

A WEB-BASED TOOL FOR CONSTRUCTING TEXT SUMMARIZATION CORPUS

Author: **Nguyen Anh Tu**
Class ICT-59

Supervisor: **Dr. Nguyen Thi Thu Trang**

HANOI, May 2019

REQUIREMENTS FOR THE THESIS

1. Student information

Student name: Nguyen Anh Tu

Tel: +84 36 458 9476 Email: tunguyen.hust@gmail.com

Class: ICT – 59 Program: Fulltime

This thesis is performed at Room 914, Ta Quang Buu Library, Hanoi University of Science and Technology.

2. Goal of the thesis:
   Create a web-based tool for constructing text summarization corpus with completed and professional process.
3. Main tasks
   - Study the requirement of the system.
   - Study and adopt ReactJs framework to create User Interface for the web-based tool to fulfill the requirement.
   - Adopt NodeJs, ExpressJs and ensure the architecture of the system follow Microservice architect.
   - Test the system carefully before release.
4. Declaration of student

I - *Nguyen Anh Tu* - hereby warrants that the Work and Presentation in this thesis are performed by myself under the supervision of *Dr. Nguyen Thi Thu Trang*. All results presented in this thesis are truthful and are not copied from any other work. All references in this thesis - including images, tables, figures, and quotes - are clearly and fully documented in the bibliography. I will take full responsibility for even one copy that violates school regulations.

Hanoi, 24 May 2019

Author

Nguyen Anh Tu

5. Attestation of the supervisor on the fulfillment of the requirements of the thesis:

Hanoi, 24 May 2019

Supervisor

Dr. Nguyen Thi Thu Trang

ACKNOWLEDGEMENT

First of all, I would like to express my sincere gratitude to my supervisor, Dr. Nguyen Thi Thu Trang, for continuous support and guidance my thesis for her patience. She not only gives me golden advice to solve the challenges of the thesis but also teaches me a lot about the way to work and organize time. Without her, it is impossible for me to do this thesis and I would like to thank her again for great support and guidance.

Second, I would like to say thank to my team who has worked with me in this graduation projects, and all of my classmates who has accompanied with me over the last five years at university.

I would like to thank all professors of Hanoi University of Science and Technology who have taught me since the day I entered the university five years ago. They are always willing to support students, and that really impresses me. Their expert knowledge inspired me a lot to study more and follow my passion.

Finally, I would like to thank my parents, who have always stood by me, taken care for me, and given me a lot of motivation during my student life so I have enough strength and confidence to follow my passion.

Hanoi, May 2019,

Nguyen Anh Tu.

TÓM TẮT

Hiện nay, với sự phát triển của khoa học công nghệ, và sự bùng nổ của Internet trong thời đại công nghiệp 4.0, ngày càng nhiều những trang thông tin điện tử, báo điện tử ra đời với một tốc độ khủng khiếp. Con người ngày nay đã có thể tiếp cận với lượng thông tin khổng lồ thông qua mạng xã hội, các trang báo điện tử hàng ngày chỉ với những cú click chuột hay thông qua thiết bị di động thông minh. Tuy nhiên, để nhận biết được đâu là những nguồn tin chính thống, nguồn tin có chất lượng, tránh đọc phải những tin giả mạo, những tin tức gây hoang mang dư luận, sai sự thật, thì chúng ta hiện chưa có giải pháp nào ngoài việc đọc và kiểm chứng. Cuộc sống hối hả làm con người cũng ít thời gian dành cho bản thân mình, vì vậy, con người cũng cần phải làm sao vừa rút ngắn được thời gian đọc mà vẫn tiếp cận được với một lượng thông tin đầy đủ, chính xác. Thấu hiểu điều đó, chúng ta đã và đang nghiên cứu, phát triển các thuật toán tóm tắt văn bản cùng với sự nghiên cứu cho học máy, học sâu.

Tuy nhiên, nghiên cứu và phát triển công nghệ tóm tắt văn bản không phải là một điều dễ dàng, nhất là đối với hệ thống văn bản tiếng Việt. Điều quan trọng đối với bất kì thuật toán học máy nào, đó là chúng ta cần một bộ ngữ liệu đủ sạch và đủ lớn để làm huấn luyện cho học máy. Vì vậy, trước hết, ta cần chuẩn bị một lượng dữ liệu lớn để làm bước khởi đầu cho công cuộc nghiên cứu sau này. Đề tài đồ án này đã ra đời với mục đích xây dựng một môi trường hoàn thiện, quy trình chuyên nghiệp và công cụ đầy đủ giúp ta xây dựng, thu thập dữ liệu để tiến hành làm bộ ngữ liệu cho việc tóm tắt văn bản tiếng Việt. Đây cũng là một trong những nhiệm vụ của đề tài nghiên cứu cấp quốc gia của Viện Công nghệ Thông tin và Truyền thông, Đại học Bách Khoa Hà Nội.

Em đã lựa chọn việc xây dựng một công cụ web sử dụng thư viện ReactJs và kiến trúc Microservices cho đề tài của mình. Công cụ web này cung cấp giao diện người dùng để hỗ trợ cho việc chuẩn bị ngữ liệu tóm tắt và các quy trình đánh giá, kiểm duyệt ngữ liệu một cách đầy đủ, đảm bảo ngữ liệu đầu ra đạt được chất lượng tốt nhất. Kết quả hiện tại trang web đã được đưa đến tay người dùng thật, sử dụng thật. Bộ Quốc phòng đã sử dụng công cụ để tiến hành gán nhãn mẫu và Viện Ngôn ngữ đã và đang sử dụng để xây dựng dữ liệu huấn luyện cho đề tài.

ABSTRACT OF THESIS

Now, with the development of science and technology, and the boom of the internet in the industrial age of 4.0, more and more websites and electronic news websites have been born at a tremendous speed. People today are able to access huge amounts of information through social networks, online newspapers with just a few clicks on computers or through smart mobile devices. However, in order to identify what is the official source of information, quality information sources, avoid reading fake news, confusing news and false news, we currently do not have any solutions other than reading and verifying. The bustling life makes people have less time for themselves, so people also need to do how to shorten the reading time and still have access to a sufficient and accurate information. Understanding that, we have been studying and developing text summarization algorithms along with research for machine learning and deep learning.

However, technology research and development for text summarization is not an easy thing, especially for Vietnamese language. It is important for any machine learning algorithm, that we need a clean set of language that is clean enough and large enough to train machine learning. Therefore, first of all, we need to prepare a large amount of data to be the first step in our future research. This project was born with the aim of building a perfect environment, professional process and complete tools to help us build and collect data to make a set of corpus for summarizing documents in Vietnamese. This is also one of the tasks of the national research topic of School of Information and Communication Technology, Hanoi University of Science and Technology.

I chose to build a web tool using the ReactJs library and the Microservices architecture for my topic. This web tool provides a user interface to support the construction of corpus and the evaluation process, full verification steps, ensuring the output of the system to achieve the best quality. Currently, website has been delivered to real users. The Ministry of Defense used the tool to conduct sample labeling and the Language Institute was using to build training data for the research.

TABLE OF CONTENTS

List of figures

List of tables

Acronym

NLP	Natural Language Processing
UI	User Interface
UX	User Experience
HTML	HyperText Markup Language
DBMS	Database Management System
DOM	Document Object Model
JSON	JavaScript Object Notation

Chapter 1 Introduction

In the first chapter, I would like to introduce the overview topic of this thesis, the reasons that led to my decision to choose the problems to be solved in my graduation research. Moreover, this part contains the purpose and the solution of the topic, as well as the introduction of the report outline.

1.1 Motivation

Throughout the period, reading has always been the indispensable part of human lives. Books, newspapers, and magazines are considered as the treasure of knowledge and information, which provide people lots of golden information and intellectuality. Besides, reading articles on newspapers and magazines is one of many methods that help us to get updated with our daily lives and get access to the dramatic change of the modern world day by day.

Nowadays, with the advancement of information and communication technology, along with the presence of Internet, there are more and more news webpages created with a tremendous rate. Just by a click on Internet, people can access to millions of articles from all categories including social life, culture, economy, education, and sport, etc. However, this has been leading to a common problem that readers are encountering: the enormous amount of information. Accompany with the bustling life, people tend not to have enough time for reading and getting information from articles. People cannot spend too much time and effort to read and understand long and complicated news, of which politics can be a good example.

Hence, the problem is how people can save the time of reading while they still can understand articles fully and not be outdated. This problem could be solved if we can make any information from news sites to become more concise. With the advent and development of machine learning, especially the Natural Language Processing (NLP), there are many advanced models designed for text summarization purpose. These current NLP models have been developed and tested to result in high accuracy output.

However, we have not have any complete models for text summarization that can be applied to Vietnamese articles. Therefore, at the very first step of building a text summarization, we need to construct a complete and high qualified corpus dataset, which will become the data for training and testing models in the future works of developing text summarization for Vietnamese language. In short, to finish this step of preparation and construction of corpus data, we need to build a system that can support us in collecting and constructing the data. Moreover, this system should have high availability to be accessed and used by many users at a time. The reason of this is we need to collect an enormous dataset, so the more number of users we have, the less time we need to spend on. This graduation thesis also proposes for one of the tasks of the national research topic of School of Information and Communication Technology, Hanoi University of Science and Technology, ordered by the Ministry of Defence.

1.2 Objectives and scope

In the context of the necessity of a reliable and large enough data source, the tool that is need will have several attributes, which needs to be advantages to support us in constructing corpus for text summarization in Vietnamese.

First, this system should be user-friendly and approachable. One of the most important aspect is the accessibility of the system. The tool needs to have well designed User Interface (UI) and improved the User Experience (UX), is compatible with different types of display and the performance is stability.

Second, this system should be operated with a professional procedure. The corpus data will affect directly to the accuracy of an NLP model, therefore, the requirement of the corpus data must be strict. To build a dataset with high quality, we need to have a professional and highly supervised procedure.

To achieve these goals, this system is going to have the following functions:

- There are 4 main features for constructing corpus for 4 types of summarization: (i) Single abstractive summarization, (ii) Multiple abstractive summarization, (iii) Single extractive summarization, and (iv) Multiple extractive summarization.
- Save and submit functions to allow users to store their work in database, which is based on user role (Annotator, Reviewer, Group manager, and Manager)
- Reviewer, group manager, and manager are allowed to rate the summarization of annotator if it is good or bad result. They also can reject annotator’s work for redo.
- Report function to mark an error article.

1.3 Approaches

Understanding the importance of the corpus data to the accuracy of summarization method as well as how many users we need to constructing the data, the question is how to make the tool accessible to users. Obviously, nowadays we access to software through two main directions: mobile applications and web applications. With the first approach, through mobile applications, not everyone has a phone with enough capacity to serve many applications. However, the content of an article is often too long to be read on a mobile screen, which would lead to difficulties to user when preparing the corpus.

Based on the difficulties of using mobile applications for this project and the requirements of this system, I decided to choose the second approach, which is building web application. Access via the website has clearly overcome most of the problems that users of mobile application encountered. Moreover, anyone who has an Internet-connected electronic device can access and use the web. The interface and performance of web application are not dependent on the device that users are using. On the developer side, we just need to focus on building a single platform, the web. Thus, the operation, repair and update are also easier and less expensive.

Next, after choosing a web application approach, the problem is how to build an interface compatible with many different types of screens and that interface needs to be friendly and improve user experience. Obviously, the web application has overcome most of the problems that a mobile application has, but the problem that web applications still need to solve is UI / UX. However, with the web application, it is not possible to limit the type of user display, as well as not knowing what kind of screen the user will use and therefore, the UI / UX design and construction will also be difficult. In order to deal with this disadvantage, I choose to design and build in a Web Responsive style. With a responsive design, we just need to add CSS snippets for each type of display. As a result, we can design and build an interface that is compatible, friendly with all kinds of screens and thereby also enhances UI / UX.

Currently, there are many frontend frameworks and libraries for web development. I decided to use ReactJS because this was built and developed by Facebook, so it should be guaranteed, long-term and stable. Besides, ReactJS helps to build web in the form of components that can be reused.

With the system mentioned above, there are quite a number of functions that need to be built, such as the functions of summarization, review or approval to ensure the output result of the system. If we build according to the traditional web architecture model, the centralized model will make all the functions on one. If a component is broken, then the entire site will stop working, and must find errors in the entire block and have to wait to re-deploy the entire system. This certainly takes a lot of time from having to find and fix bugs in a multi-component system, as well as taking a lot of time to re-run the system.

To solve this problem, I decided to choose modern Microservice web architecture model. In particular, components related to each other will be gathered as a traditional block model, these blocks will operate independently and depend on each other in the least possible way. In particular, components related to each other will be gathered as a traditional block model, these blocks will operate independently and depend on each other in the least way. If a block does not work, it will be easier to find, fix and re-deploy because it only needs to work on part of the system instead of the whole system as before. At the same time, if you want to expand the function, just add another block.

Finally, to make the output result of the system qualified and reliable, I decided the system will have four main roles of users: Annotator, Reviewer, Group Manager, and Manager. Annotator will be the person who directly summarize the article while the rest will be in charge of review and approval to ensure the summarization process of Annotator has the right and good enough result. This is a special process that I proposed to be implemented in the system in order to assure the quality of the corpus.

1.4 Structure of thesis

The rest of this graduation thesis is organized as follows.

Chapter 2 demonstrates the analysis of current situation and the overview of requirements. This chapter contains the illustration of all use cases of the system by using use case diagrams, details of main professional process and the use case specification.

Next, in Chapter 3, I would like to tell about the technology that I have used in my project. This includes the programming language and framework applied for frontend, which is HTML, Javascript and ReactJs. On the backend side, there is an introduction about NodeJs and ExpressJs along with the NoSQL database.

Chapter 4 will explain everything about the design of the system. In this chapter, I would like to give the overview of the architecture design and the UI design of the tool. In addition, database design would be illustrated.

Chapter 5 contains my outstanding contribution after the research. Besides, all of the difficults and troubles I faced during the progress of researching and developing this system would be included.

Finally, in Chapter 6, there is a conclusion of the thesis, which would be about the result after this project. Moreover, I would like to tell about the orientation of the future works for this system.

Chapter 2 Requirements analysis

This chapter will illustrate everything about requirements analysis. I would like to show about the general use cases of the system, then there are the decomposition diagrams of every general use cases and the activity diagram presents for the business process of the system. Finally, some main use cases and functions are chosen to be specified in order to help readers to understand about the tool.

2.1 Context

My system is a part that contributes in a large scale system of a national research topic ordered by the Ministry of Defence, which is "Automatic crawling, multi-label classification, relation extraction, extractive and abstractive summarization for single and multiple documents of Vietnamese news from Internet".

In the era of information technology, any individuals, organizations and countries all over the world have thoroughly used all means of communication to implement their intentions, guidelines and plans. Especially in the context of the industrial revolution 4.0, the Internet has become indispensable for humans. Digital media in the Internet environment increasingly plays an important role to dominate all areas of social life. Almost every activities of human have been being recorded and transformed to news, posts, or articles on social networks and online news. Therefore, many government agencies, security and intelligence organizations and technology firms in many countries have invested in building professional models and solutions that use technical facilities to collect and analyze news from sources on the Internet to detect new signs, moves and forecast movement and development trends of things and phenomena.

2.2 Feature overview

2.2.1 General use case diagram

In this system, there are 4 actors: (i) Annotator, (ii) Reviewer, (iii) Group Manager, and (iv) Manager. Any actor can manage their tasks with use case "Request tasks" and "View task list" by extending the User actor.

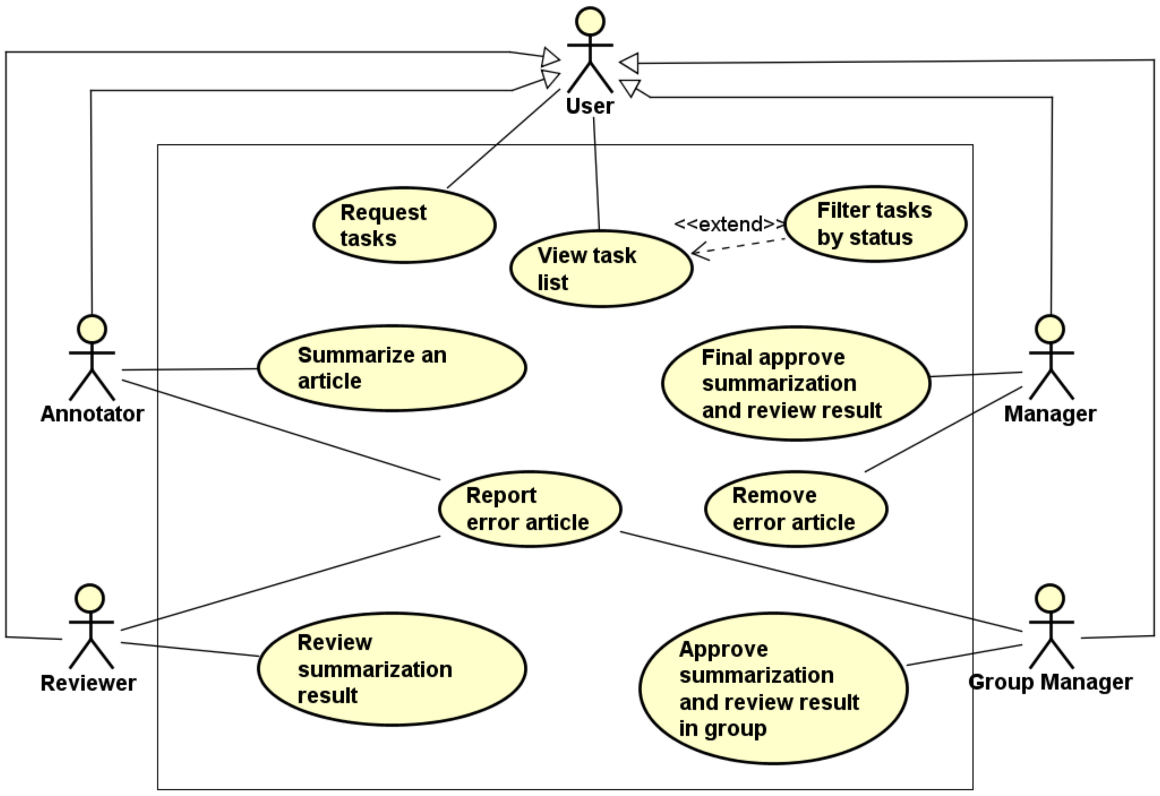


Figure 1: General use case diagram.

Annotator is the person who do the very first step in the whole process of constructing corpus for text summarization. Their main action is creating summary for each article which is assigned to them. This job is included in the use case "Summarize an article".

After Annotators have finished their job, articles are assigned to Reviewer to do the next process, which is "Review summarization result". In this use case, Reviewer has to check carefully the quality of the summary that composed by Annotator, then decide to submit the result to move to next step, which is Group Manager's responsibility.

Group Manager and Manager will have the role to approve a summarization result. The summary is moved to Group Manager to be approved after reviewed by Reviewer. The approval process will happen twice, the first is by Group Manager and then Manager will take responsibility. The use case "Final approve summarization and review result" is the last use case in the whole process of the system.

Annotator, Reviewer, and Group Manager can report an error article, then this article will be directly moved to Manager tasks to be removed.

2.2.2 Use case decomposition diagram

2.2.2.1 Use case "Summarize an article" decomposition

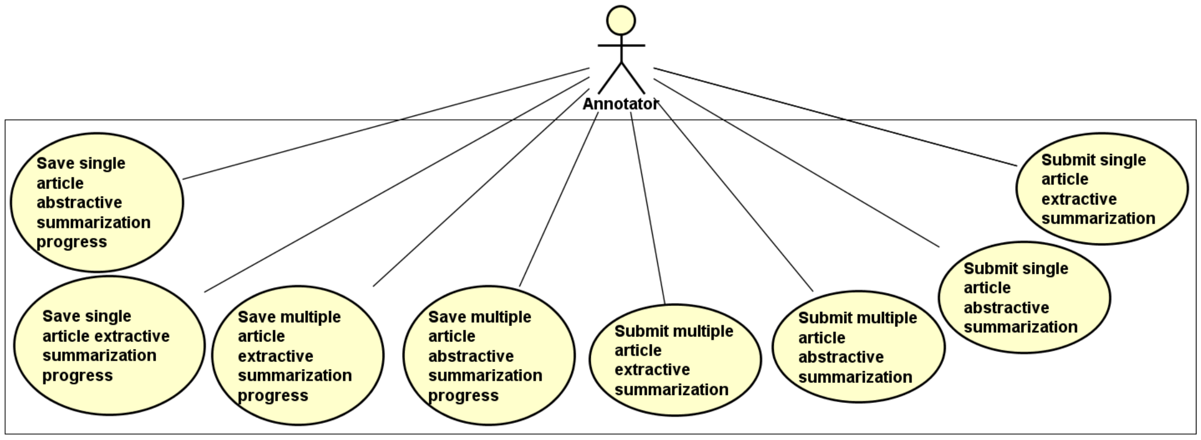
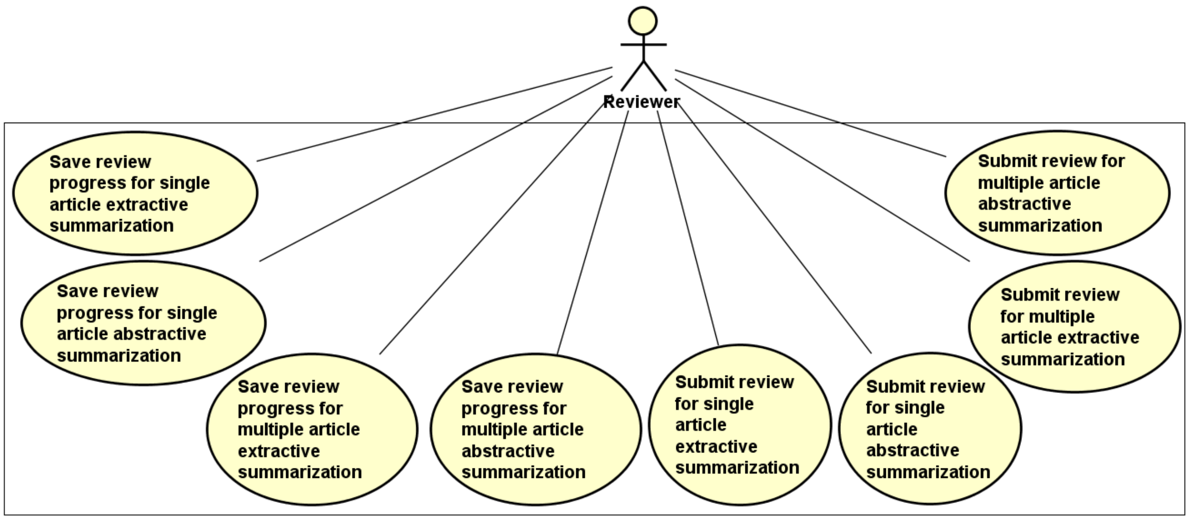

Annotator
Save single article abstractive summarization progress
Save single article extractive summarization progress
Save multiple article extractive summarization progress
Save multiple article abstractive summarization progress
Submit multiple article extractive summarization
Submit multiple article abstractive summarization
Submit single article abstractive summarization
Submit single article extractive summarization

Figure 2: Use case "Summarize an article" decomposition diagram.

In the process of summarizing an article, Annotator can work with four types of summarization: (i) Single article abstractive summarization, (ii) Multiple article abstractive summarization, (iii) Single article extractive summarization, and (iv) Multiple article extractive summarization. In each type of summarization, there are two use cases "Save" and "Submit". "Save the summarization progress" use cases will help Annotator to store the current progress of their works in any type of summarization, in order not to lose progress data if they want to back to the previous page or jump to another task. After a careful summarization process, Annotator can send the official result, which includes the summary and comments about the task (if any), to the system by submission. Please notice that there is no change can be made after submission, which is use case "Submit the summarization result".

2.2.2.2 Use case "Review summarization result" decomposition

Same as Annotators, Reviewers have to work with four types of text summarization. This is the second step in the summarization process after annotator has done. In this use case, reviewer is the first role to check the annotator's summarization work. Reviewer not only can update the summary made by annotator but also can comment and give the scorer for annotator's work.

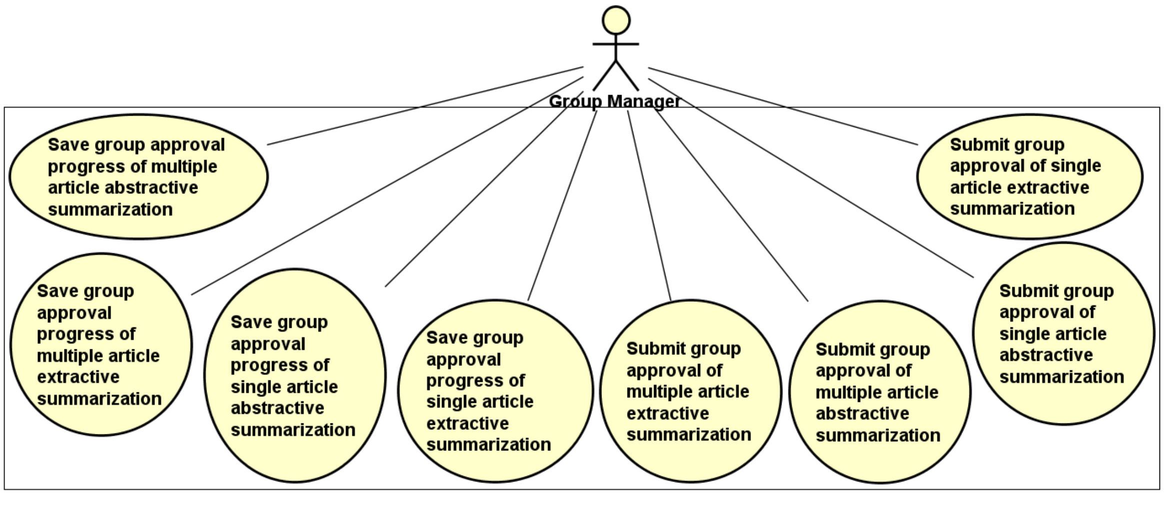


Reviewer
Save review progress for single article extractive summarization
Save review progress for single article abstractive summarization
Save review progress for multiple article extractive summarization
Save review progress for multiple article abstractive summarization
Submit review for single article extractive summarization
Submit review for single article abstractive summarization
Submit review for multiple article extractive summarization
Submit review for multiple article abstractive summarization

Figure 3: Use case "Review summarization result" diagram.

Reviewers have to read carefully the article and its summary before grading the result. Each task in each summarization type of a reviewer will contain the reading, comment, and giving score for the summarization result of Annotator. They also can save the work progress and edit that later. But after submitting the review, which includes the summary update (if any), comments, and grading, they cannot edit anymore.

2.2.2.3 Use case "Approve summarization and review result in group" decomposition



Group Manager
Save group approval progress of multiple article abstractive summarization
Save group approval progress of multiple article extractive summarization
Save group approval progress of single article abstractive summarization
Save group approval progress of single article extractive summarization
Submit group approval of multiple article extractive summarization
Submit group approval of multiple article abstractive summarization
Submit group approval of single article abstractive summarization
Submit group approval of single article extractive summarization

Figure 4: Use case "Approve summarization and review result in group" diagram.

Group Manager in charge of approving the review and the summarization result of four types of text summarization. This will happen after reviewer has finish the task.

This actor also can save the work progress during approval, which is use case “Save the group approval progress” in the above diagrams. Group Manager needs to provide the grade before submitting the result. Approval submission will change status of the task to the last process, which is responsibility of Manager. No change can be made after submission.

2.2.2.4 Use case “Final approve summarization and review result” decomposition

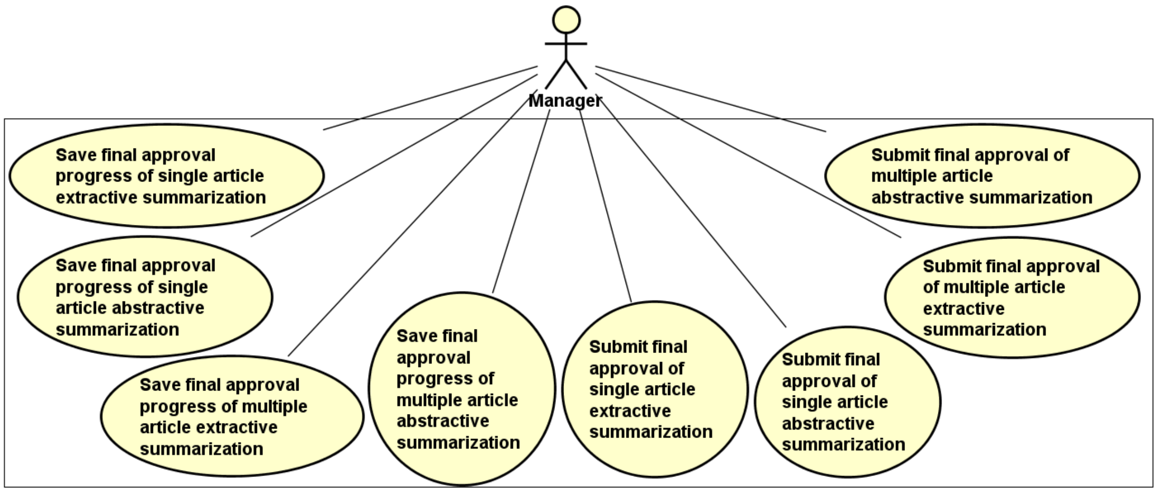

Manager
Save final approval progress of single article extractive summarization
Save final approval progress of single article abstractive summarization
Save final approval progress of multiple article extractive summarization
Save final approval progress of multiple article abstractive summarization
Submit final approval of single article extractive summarization
Submit final approval of single article abstractive summarization
Submit final approval of multiple article extractive summarization
Submit final approval of multiple article abstractive summarization

Figure 5: Use case “Final approve summarization and review result” diagram.

Final approve is responsibility of Manager, which is the highest role in this system. This role will be in charge of final approval only. This step is the last step of the whole process of text summarization in four practices.

It can be seen from Figure 5 that this use case has 8 details use case for 4 types of summarization and for action “Save” and “Submit”. Managers also can store their approval progress during their work by saving them. However, Managers have to check carefully the review and summarization result because their task is final approve which decide if the summarization is accepted and can be used as corpus. Submitting the final approval also store the comment and the score given by Manager to Annotator. The above diagrams for managing in four types of text summarization have shown the use cases for “Save” and “Submit”.

2.2.3 Business process

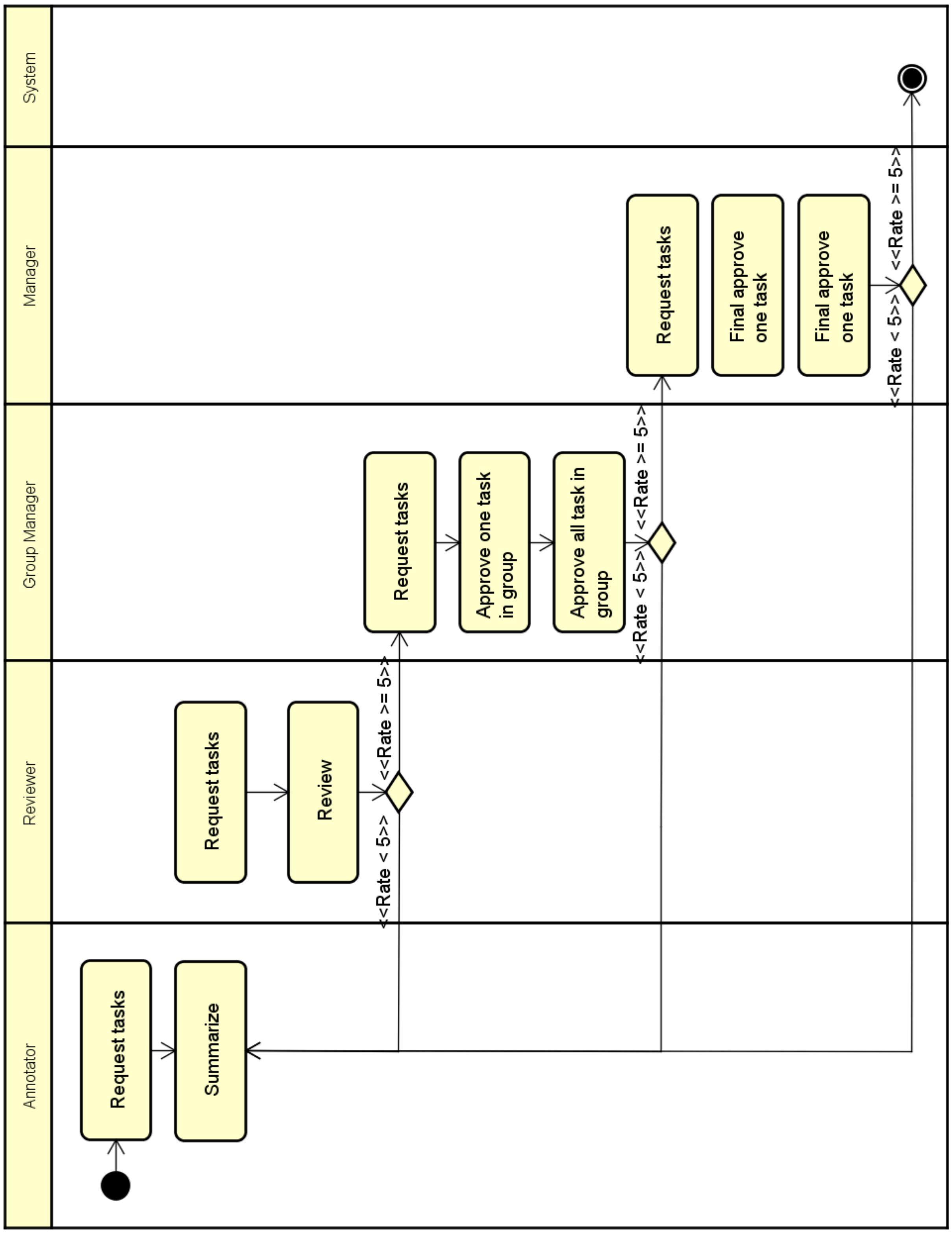


Figure 6: Activity diagram of complete summarization process.

Any authenticated user can request task for their roles. To start the process, user clicks the “Lấy văn bản” (Request tasks) button, then system will show a confirmation modal with a message to inform how many tasks the user can get at once. Then the system will check if there are unassigned tasks. If no tasks can be assigned, system will display an error message, else it will automatically assign tasks to user and display the result.

Annotator is the person who in charge of “Summarization” process. In the list of assigned tasks, annotator can choose any task to start working. System will direct to the page that show details of a task, which contains article’s content. If the article’s data is wrong or lost, annotator can report this by clicking “Báo lỗi” (Report) button, this will make the system change the status of the article as well as the task to “Lỗi”. If everything is normal, annotator can start summarizing the article by entering the summary and comments (if any). “Lưu” (Save) button helps annotator to store the work progress. When annotator have finished the task, click “Gửi kết quả” (Submit) button and confirm the submit to upload the result. After submission, no changes can be made. System will display the submitted work in “Lịch sử” (History).

Reviewer can review the summarization result of an annotator. In the list of assigned tasks, reviewer can choose any task to start working. System will direct to the page that show details of a task, which contains article’s content, summarization result from annotator, and comments of annotator. If the article’s data is wrong or lost, annotator can report this by clicking “Báo lỗi” (Report) button, this will make the system change the status of the article as well as the task to “Lỗi” (Error). If everything is normal, annotator can start reviewing by reading carefully the summary and check if that summary is appropriate with the article. Then reviewer can grade the summarization result (in the range of 1-10). “Lưu” (Save) button helps reviewer to store the work progress. When reviewer have finished the task, click “Gửi kết quả” (Submit) button and confirm the submit to upload the review. After submission, no changes can be made. System will display the submitted work in “Lịch sử” (History).

Then the article status will be changed to “Chờ quản lý nhóm phê duyệt” (Ready for group manager approve). Now task will be displayed in the list of “Quản lý nhóm” (Group manage) tab. Group manager can choose any task to start working and system will direct to the page that show details of a task. If the article’s data is wrong or lost, group manager can report this by clicking “Báo lỗi” (Report) button, this will make the system change the status of the article as well as the task to “Lỗi” (Error). If everything is good, group manager can grade the result (in the range of 1-10) then click “Lưu” (Save) button or “Gửi kết quả” (Submit) button to store the work as in Annotator and Reviewer’s parts. Now the status will be changed to “Chờ phê duyệt” (Ready for approve).

Manager will be assigned to that task. Manager can choose any task to start working and system will direct to the page that show details of a task. If the task status is "Lỗi" (Error), manager have to check if it is error or not, then click "Loại bỏ" (Remove) button to remove the article. If everything is good, manager can grade the result (in the range of 1-10) then click "Lưu" (Save) button or "Gửi kết quả" (Submit) button to store the work. Until then, the process is complete.

2.3 Feature description

The following table is the list of all use cases of the system.

Table 1: List of use cases

General use case	Use case code	Use case name
	UC001	Request tasks
	UC002	View task list
	UC003	Filter task by status
Summarize an article	UC004	Save single article extractive summarization progress
	UC005	Submit single article extractive summarization
	UC006	Save multiple article extractive summarization progress
	UC007	Submit multiple article extractive summarization
	UC008	Save single article abstractive summarization progress
	UC009	Submit single article abstractive summarization
	UC010	Save multiple article abstractive summarization progress
	UC011	Submit multiple article abstractive summarization
Review summarization result	UC012	Save review progress of single article extractive summarization
	UC013	Submit review for single article extractive summarization
	UC014	Save review progress of multiple article extractive summarization
	UC015	Submit review for multiple article extractive summarization
	UC016	Save review progress of single article abstractive summarization

	UC017	Submit review for single article abstractive summarization
	UC018	Save review progress of multiple article abstractive summarization
	UC019	Submit review for multiple article abstractive summarization
Approve summarization and review result in group	UC020	Save group approval progress of single article extractive summarization
	UC021	Submit group approval of single article extractive summarization
	UC022	Save group approval progress of multiple article extractive summarization
	UC023	Submit group approval of multiple article extractive summarization
	UC024	Save group approval progress of single article abstractive summarization
	UC025	Submit group approval of single article abstractive summarization
	UC026	Save group approval progress of multiple article abstractive summarization
	UC027	Submit group approval of multiple article abstractive summarization
Final approve summarization and review result	UC027	Save final approval progress of single article extractive summarization
	UC028	Submit final approval of single article extractive summarization
	UC029	Save final approval progress of multiple article extractive summarization
	UC030	Submit final approval of multiple article extractive summarization
	UC031	Save final approval progress of single article abstractive summarization
	UC032	Submit final approval of single article abstractive summarization
	UC033	Save final approval progress of multiple article abstractive summarization
Final approve summarization and review result	UC034	Submit final approval of multiple article abstractive summarization
	UC035	Report error article
	UC036	Remove error article

In the following part, there is specification of use cases of the system. Because of the limitation of the thesis, I would like to specify some main functions only.

2.3.1 Use case “Submit single article extractive summarization”

Table 2: Use case “Submit single article extractive summarization” specification

Use case code	UC005	**Use case name**	Submit single article extractive summarization
Actor	Annotator		
Pre-condition	Exist assigned task for annotator		
Main flow of events (Success)	**No**	**Actor**	**Action**
	1.	Annotator	Select tab “Gán nhãn” (Annotate)
	2.	System	Display list of assigned tasks
	3.	Annotator	Select a task
	4.	System	Display details of task: article’s content
	5.	Annotator	Select sentence to put in summary
	6.	Annotator	Re-order sentence and comment
	7.	Annotator	Submit the summarization result
	8.	System	Display the confirmation message
	9.	Annotator	Confirm to submit
	10.	System	Upload the result
	11.	System	Display success message
	12.	System	Display Annotator’s work in History
Alternative flow of events	**No**	**Actor**	**Action**
	9a.	Annotator	Cancel the submission
	10a.	System	Display details of task: article’s content and current work progress
Post-condition	None		

Input data of summary include these data fields:

No	**Data fields**	**Description**	**Mandatory**	**Valid condition**	**Example**
1.	Summary	The brief content of article	Yes		This is summary.
2.	Comment	Comment task	No		Good article.

2.3.2 Use case “Submit single article abstractive summarization”

Table 3: Specification of “Submit single article abstractive summarization”

Use case code	UC009	**Use case name**	Submit single article abstractive summarization
Actor	Annotator		
Pre-condition	Exist assigned task for annotator		
Main flow of events (Success)	**No**	**Actor**	**Action**
	1.	Annotator	Select tab “Gán nhãn” (Annotate)
	2.	System	Display list of assigned tasks

	3.	Annotator	Select a task
	4.	System	Display details of task: article's content
	5.	Annotator	Type the summary and comment
	6.	Annotator	Submit the summarization result
	7.	System	Display the confirmation message
	8.	Annotator	Confirm to submit
	9.	System	Upload the result
	10.	System	Display success message
	11.	System	Display Annotator's work in History
Alternative flow of events	**No**	**Actor**	**Action**
	8a.	Annotator	Cancel the submission
	9a.	System	Display details of task: article's content and current work progress
Post-condition	None		

Input data of summary include these data fields:

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Summary	The brief content of article	Yes		This is summary.
2.	Comment	Comment task	No		Good article.

2.3.3 Use case "Submit review for single article extractive summarization"

Table 4: Specification of "Submit review for single article extractive summarization"

Use case code	UC013	**Use case name**	Submit review of single article extractive summarization
Actor	Reviewer		
Pre-condition	Exist assigned task for reviewer		

Main flow of events (Success)	**No**	**Actor**	**Action**
	1.	Reviewer	Select tab "Đánh giá" (Review)
	2.	System	Display list of assigned tasks
	3.	Reviewer	Select a task
	4.	System	Display details of task: article's content, annotator's summary
	5.	Reviewer	Check the summary and update (if needed)
	6.	Reviewer	Grade the summary and comment
	7.	Reviewer	Submit the summarization review result
	8.	System	Display the confirmation message
	9.	Reviewer	Confirm to submit
	10.	System	Upload the result
	11.	System	Display success message
	12.	System	Display Annotator and Reviewer's work in History

Alternative flow of events	**No**	**Actor**	**Action**
	9a.	Reviewer	Cancel the submission
	10a.	System	Display details of task: article’s content and current work progress
Post-condition	None		

Input data of summary include this data fields:

No	Data fields	Description	Mandatory	Valid condition	Example
1.	Summary	The brief content of article	Yes		This is summary.
2.	Comment	Comment about the article	No		Good article.
3.	Grade	Score of annotator	Yes	Integer number (1-10)	8

2.3.4 Use case “Report error article”

Table 5: Specification of “Report error article”

Use case code	UC035	**Use case name**	Report error article
Actor	User (Annotator / Reviewer / Group Manager)		
Pre-condition	Exist assigned task for user		
Main flow of events (Success)	**No**	**Actor**	**Action**
	1.	User	Select tab “Gán nhãn / Đánh giá / Phê duyệt nhóm” (Annotate / Review / Group manage)
	2.	System	Display list of assigned tasks
	3.	User	Select a task
	4.	System	Display details of task
	5.	User	Check details of task: article’s title and content
	6.	User	Click “Báo lỗi” (Report)
	7.	System	Display the confirmation message
	8.	User	Confirm to report
	9.	System	Change status of task to “Lỗi” (Error)
	10.	System	Display success message
Alternative flow of events	**No**	**Actor**	**Action**
	8a.	User	Cancel the report
	9a.	System	Display details of task
Post-condition	None		

2.3.5 Use case “Remove error article”

Table 6: Specification of “Remove error article”

Use case code	UC036	**Use case name**	Remove error article
Actor	Manager		
Pre-condition	Exist assigned task with “Error” status for manager		
Main flow of events (Success)	**No**	**Actor**	**Action**
	1.	Manager	Select tab “Phê duyệt” (Manage)
	2.	System	Display list of assigned tasks
	3.	Manager	Select a task with status “Lỗi” (Error)
	4.	System	Display details of task
	5.	Manager	Check details of task: article’s title and content
	6.	Manager	Click “Loại bỏ” (Remove)
	7.	System	Display the confirmation message
	8.	Manager	Confirm to remove
	9.	System	Change status of task to “Đã hủy” (Removed)
	10.	System	Display success message
Alternative flow of events	**No**	**Actor**	**Action**
	8a.	Manager	Cancel the remove
	9a.	System	Display details of task
Post-condition	None		

Chapter 3 Adopted technology and framework

As mentioned in chapter Chapter 1, this system is constructed in the web-based and contains 2 main parts: Frontend and Backend. In each part, specific programming language, architecture and technology will be applied to approach the solution.

3.1 Programming languages and technologies used in Front-end

3.1.1 Basic technology for web programming

3.1.1.1 HTML and HTML5

HTML (Hyper Text Markup Language) is the standard markup language designed for creating web pages and web applications. HTML is one of the three main elements used to create a website and it is the only dispensable element.

HTML5 is the latest version of HTML, designed to resolve and improve some disadvantages of HTML. When using HTML5, the internet browser does not have to install any add-ons or plugins to display the website. Many new features and code syntaxes is added in HTML5.

3.1.1.2 CSS

CSS (Cascading Style Sheet) supports the web browser to display the HTML content. It could be understood that CSS is a language that describes the style of an HTML document. It makes the website to become colored and more attractive. In CSS, we can control the font style, font size, color, and many attributes of the website.

3.1.1.3 Web Responsive and Bootstrap

a) Web Responsive

Web Responsive is a design so that the website can fit any screen resolution. There are many types of screens with different sizes and resolutions, so designing the interface for just one type of screen will make the layout of the site cluttered, the size of the font is small and unreasonable when viewed on different displays. To overcome this, many developers choose to design the interface for desktop and mobile separately. However, this design unintentionally creates inadequacies, every time we want to change a certain part of the interface, we have to go to change the code in many different files, and the code will also be repeated a lot. In addition, this design will reduce the site's search result rankings because we display the same content on two different paths.

With web responsive, we can simply apply the “media” query to any screens that we want to change the UI, the rest of the code will still be applied to all other screen types. Thus, when a change occurs, we only update in one part, and therefore the source code management is also easier. Moreover, this approach also contributes to the ranking of search results for the website.

b) Bootstrap

Bootstrap is a front-end framework for designing websites and webapps. It contains pre-built and popular HTML and CSS components such as form, button, navbar, and a number of JavaScript interactions built into it. Bootstrap makes frontend programming faster because it does not have to rewrite all the common interface elements. At the same time, it also supports very well in building web responsive due to its column division system, which is bootstrap grid.

3.1.2 React and NextJs framework

3.1.2.1 React

React is a Javascript library, built and developed by Facebook for building user interfaces. The advantage of React is that it is not as heavy as other frontend frameworks, whereas it is relatively compact and easy to coordinate with other javascript libraries. Here are some of the React features: (i) Declarative Programming, (ii) Virtual DOM, (iii) JSX, (iv) Component-based, and (v) Learn Once, Write Everywhere.

Declarative Programming is shown by that the code in React describes the result you want to happen, and how to create it will be done by the computer and the compiler. This programming style helps to shorten thinking to create user interfaces. The design becomes simpler because you only have to design each part of the application and show the corresponding conditions, constraints and interfaces you want to display. React will automatically update the exact changes that correspond to the changed components when the conversion data is noticed. The source code thus also becomes easier to find errors when it occurs and the source code reading to guess the events, the interface is easier to display.

To make the interface switch only happen in components that actually change, React uses Virtual DOM. Today, the more webapp becomes more complex, the more complicated the DOM tree is. Searching and editing in the real DOM tree will reduce the performance of the website, especially when changes take place at high frequency. Virtual DOM was born to solve this problem. Virtual DOM can be considered as a copy of HTML DOM and React can easily calculate changes on it quickly, then update to the real DOM tree. As a result, the performance of the website as well as user experience increased significantly.

JSX, which is understood as Javascript XML, is a language that allows insert HTML in Javascript. JSX performs optimizations while compiling into Javascript code, errors will also be detected during compilation, thereby making debugging easier. JSX is based on JavaScript, so it is possible to apply the good features of JavaScript to write JSX.

React is built on components. Each component expresses a certain part of the entire UI of the site, and can be reused in many places without having to copy the source code when you want to use it elsewhere. Thanks to the division into components, the development and bug fixing is easier because we only have to focus on one component instead of the entire source code of the website as before. Each component includes props and state. Props help components can interact with each other, transfer data to each other; while state describes the internal state of the component, when the state changes, the component changes and the interface is updated. The flow of data in React works in a one-way mechanism, data will only be transmitted from the parent component to the child component via props.

“Learn Once, Write Everywhere.” Just as the idea shows, with React learning, we can apply the knowledge we have learned for both website and application building. React works independently with the backend, so the backend is fully applicable to building both web and app.

3.1.2.2 Redux

Unlike other MVC frameworks, React is just a library for building user interfaces so it is relatively lightweight and easily combined with other libraries. The ecosystem that React created is extremely diverse to maximize its capabilities, including react-redux that bridges Redux to manage states; react-router to manage and navigate routes; webpack for packaging applications; flux and redux to manage the status of components more easily.

Among the libraries in React's ecosystem, Redux is one of the most useful. Redux is a library used to manage the state of JavaScript applications. SPA (Single page application) with requirements are becoming more and more complex, leading to the need to manage their status. These states can be data taken from servers, static data. Contemporarily, we must also manage the state of user interfaces, such as routes, selected tabs and pagination. Managing changes of these states is not easy. In particular, we know that React works with a one-way data mechanism, so data can only pass from parent to child. Suppose that some child component wants to change the state of an ancestor component, then we will have to transmit data through a series of intermediate components, which is very useless and difficult to manage when the number of threads is on an overlap and increases.

3.1.2.3 NextJs

NextJs is a React framework, which is designed for building a web application with React but performing SSR (Server-Side Rendering). "SSR is the traditional rendering method, basically all of your page's resources are housed on the server. Then, when the page is requested, the HTML is delivered to the browser and rendered, JS and CSS downloaded, and final render appears to the user."[1] Therefore, NextJs, with SSR, help us to load pages faster initially and not worry about routing.

There are some features that makes NextJs become superior: (i) Server-rendered by default, (ii) automatic code splitting, (iii) simple client-side routing, (iv) webpack-based dev environment, (v) ability to implement with Node.js, and (vi) customize with your own Babel and Webpack configs.[2]

3.2 Programming languages and technologies used in Back-end

3.2.1 NodeJs and ExpressJs

Node.js is a server-side platform built on Google Chrome's JavaScript Engine, it is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Express.js is the most popular framework of Node.js used for building web applications and APIs. With Express.js, application development is greatly shortened. The main features of Express.js can be seen including this is a relatively compact and flexible framework, providing many powerful features on the web and mobile platforms. It provides a multitude of HTTP methods and middleware, thus making it easier and faster to build APIs. Express creates a thin layer of basic web application features without obscuring the useful features of Node.js. At the same time, many other popular frameworks are also built on Express platform such as MEAN, Sails, KeystoneJS.

3.2.2 RestAPI

In order for services in Microservice to communicate with each other, communicate with front-end and with third-party applications, web APIs are a relatively popular and widely applied solution. Web APIs perform data exchange in the form of XML, JSON, in which JSON is still the most popular form.

REST is acronym for Representational State Transfer. There are 6 main principles of REST: (i) Client-server, (ii) Stateless, (iii) Cacheable, (iv) Uniform interface, (v) Layered system, and (vi) Code on demand.[3]

With Client-server principle, the uniform interface separates clients from servers, which means clients are not concerned with data storage. Hence, it can improve the protability of the user interface across multiple platforms, server can be simpler and more scalable.

Each request from client to server must contain all of the information necessary to understand the request, and cannot take advantage of any stored context on the server. Session state is therefore kept entirely on the client. Essentially, what this means is that the necessary state to handle the request is contained within the request itself, whether as part of the URI, query-string parameters, body, or headers. The URI uniquely identifies the resource and the body contains the state (or state change) of that resource. Then after the server does it's processing, the appropriate state, or the piece(s) of state that matter, are communicated back to the client via headers, status and response body.

Cacheable means cache constraints require that the data within a response to a request be implicitly or explicitly labeled as cacheable or non-cacheable. If a response is cacheable, then a client cache is given the right to reuse that response data for later, equivalent requests.

By applying the software engineering principle of generality to the component interface, the overall system architecture is simplified and the visibility of interactions is improved. In order to obtain a uniform interface, multiple architectural constraints are needed to guide the behavior of components. REST is defined by four interface constraints: identification of resources; manipulation of resources through representations; self-descriptive messages; and, hypermedia as the engine of application state.

The layered system style allows an architecture to be composed of hierarchical layers by constraining component behavior such that each component cannot “see” beyond the immediate layer with which they are interacting.

REST allows client functionality to be extended by downloading and executing code in the form of applets or scripts. This simplifies clients by reducing the number of features required to be pre-implemented.

3.2.3 MongoDB

MongoDB is an open source database management system, a NoSQL database and is used by millions of people. MongoDB is a document-oriented database. The data is stored in document type JSON instead of tabular format as relational database so the query will be very fast. With relational database we have the concept of tables, relational databases (like MySQL or SQL Server ...) using tables to store data, then with MongoDB we will use the concept of collection instead of tables. Compared to Relational DBMS, in MongoDB, collection is like table, and document is corresponding with row. MongoDB will use documents instead of row in RDBMS. MongoDB collections are structured flexibly, allowing data to not follow a certain structure. Relevant information is stored together for quick query access through MongoDB query language.

3.2.4 Microservice architecture

Currently, most applications are built as monolithic blocks. At a small and medium level, such applications are easy to write, easy to test and deploy. However, as the number of users, the number of requests increases, the development in a single block results in making the application more cumbersome and complex. If an error arises, it is difficult to determine where the error is, and must conduct testing on millions of lines of source code. When the error was found, it took relatively long time for the deployment to resume operation. This both causes waste of time, human resources and money. In addition, block-based applications often only apply one or several languages, technology, after a time when the application blooms, it is difficult to switch because the change leads to the entire kernel. The force must update the language, new technology, and changing millions of lines of source code is obviously not easy, and not necessarily successful.

Understanding the disadvantages of the traditional monolithic model, microservice was designed for splitting large applications into interconnected services. Each service will perform a specific set of functions.

About the detail information of microservice and how it is applied in this project, I would like to present in Chapter 4 and Chapter 5.

Chapter 4 Application development and deployment

Chapter 4 delves into the design and implementation for the tasks set by the system, including the overview architecture of the system, design of components, database, UI, and application construction.

4.1 Overview architecture

4.1.1 General architecture

This figure illustrates the system architecture. It can be seen from the figure that system is divided into two main parts: Front-end and Back-end (or Client-side and Server-side).

Frontend is the interface of the system, including the interface for users to provide functions related to task management, 4 types of summarization, review, and approval.

Backend is designed according to Microservices architecture model, which includes 4 main services corresponding to 4 types of summarization: Single Extractive Summarization service, Multiple Extractive Summarization service, Single Abstractive Summarization service, and Multiple Abstractive Summarization service. These service provide functions for requesting tasks, saving and submitting summarization results. Moreover, my system also has User service, which is for user profile management, and Article service for managing all articles, including get article's content, categories and all related information. However, these service is created and implemented by other students in my group, therefore, I will not mention them in the upcoming parts.

Frontend does not connect directly to Backend services but connects via a gateway. The parts in the system communicate with each other by calling APIs, which is known as REST API as mentioned in Chapter 3.

The overview of architecture design is shown in Figure 7.

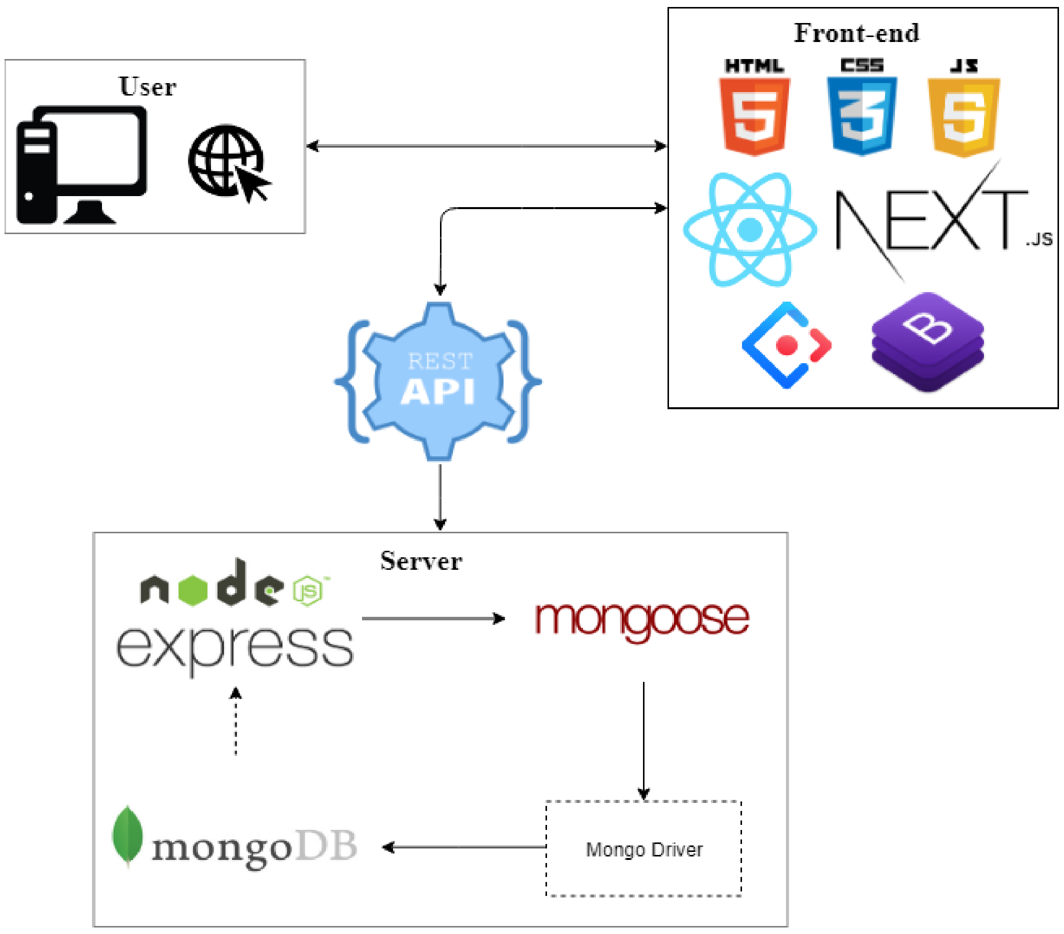


Figure 7: Overview architecture of system.

4.1.2 Frontend architecture

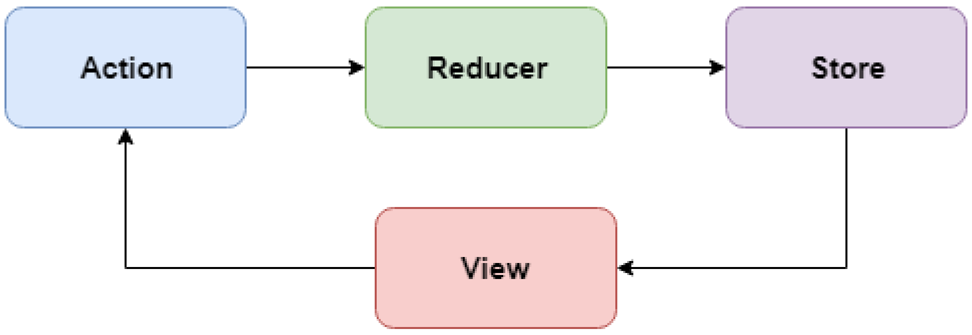


Figure 8: Frontend architecture.

Figure 8 presents the architecture for the client side of the system. Frontend consists of 4 main parts: Action, Reducer, and Store are components of Redux, and View represents React. As mentioned in Chapter 3 about why we need Redux, now I would like to present about its architecture.

Redux includes a store, actions and reducers. The states of the components in React will be registered to connect directly to the store. Whenever a state changes, instead of passing through multiple intermediaries, it will only change the store, and the registered components with the store will recognize this change and update themselves. In addition, with Redux, we can also cache data from the server, thereby increasing performance, minimizing the need to submit requests to retrieve data.

Redux is like a design, and is described with three main principles: (i) The state of the application is stored in a single object tree called "Store", (ii) the states are only allowed to read, and cannot be changed, (iii) the change is only made in pure function.

4.1.3 Backend architecture - Microservice

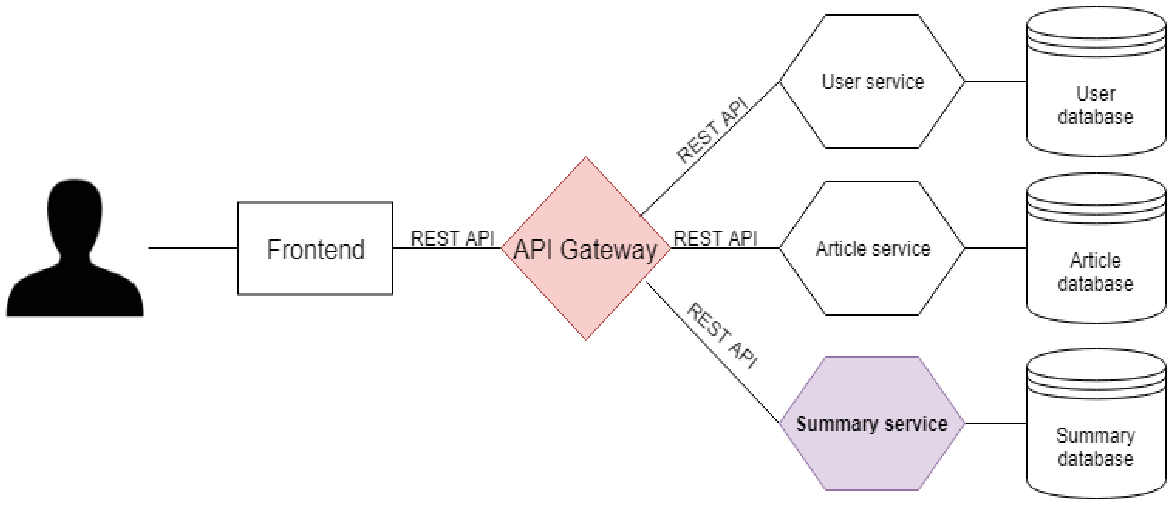


Figure 9: Backend microservice architecture.

In this system, the server side contains 3 services that serve user profile management, articles, and summarization. The user service and article service were created and implemented by another student in my group. Hence I would like to present about Summary Service only.

The summary service provides REST APIs for request articles, save work progress, submit work progress, get task details of user, and get article details (include article's title, content, categories). Summary service is applied with Express.js framework.

The components of the summary service include the Routes to receive requests and navigation, the Controllers receive requests and process them. Processing at Controllers is mostly data query operations which works with models in database.

4.2 Architecture design

4.2.1 Use case design

a) Use case "Submit the single article abstractive summarization result"

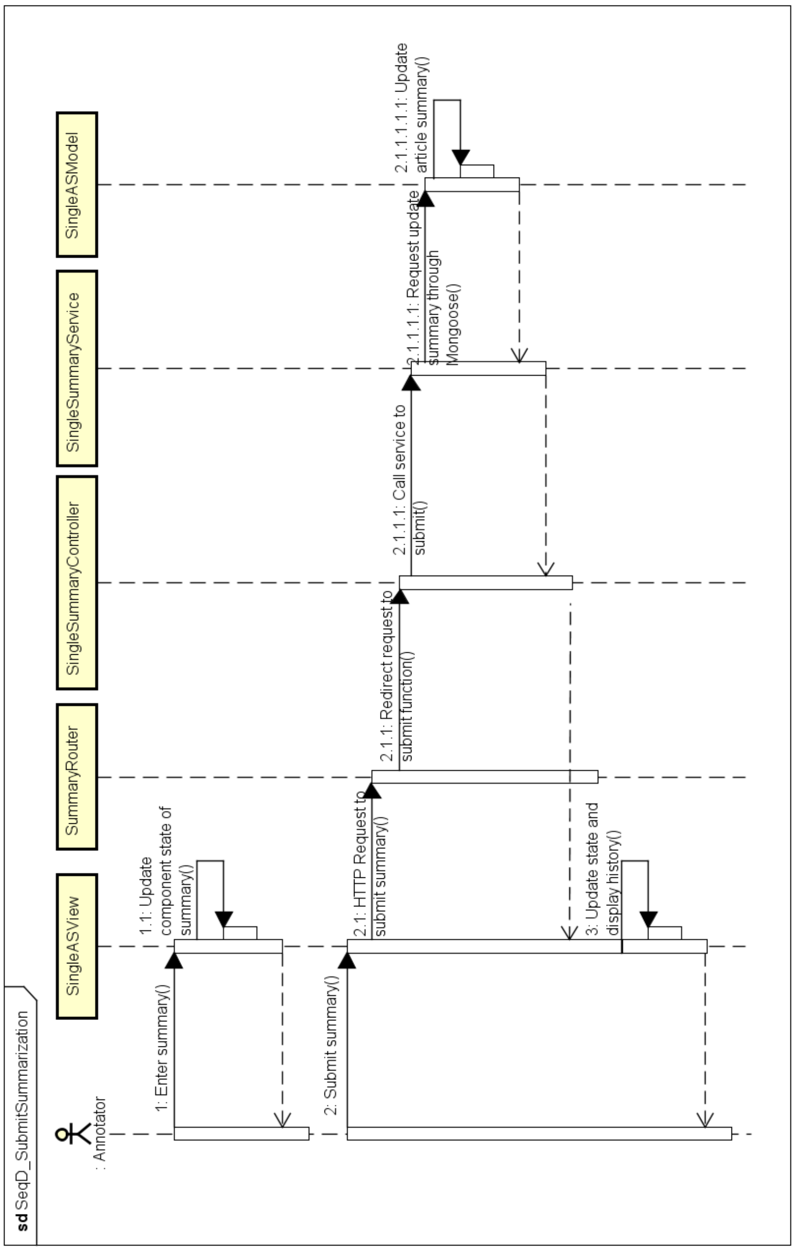

sd SeqD_SubmitSummarization
: Annotator
SingleASView
SummaryRouter
SingleSummaryController
SingleSummaryService
SingleASModel
1: Enter summary()
1.1: Update component state of summary()
2: Submit summary()
2.1: HTTP Request to submit summary()
2.1.1: Redirect request to submit function()
2.1.1.1: Call service to submit()
2.1.1.1.1: Request update summary through Mongoose()
2.1.1.1.1.1: Update article summary()
3: Update state and display history()

Figure 10: Sequence diagram of "Submit single article abstractive summarization".

This use case is one of the main functions in this system. Because of the limitation of the thesis, I would like to present this use case only, which is for the single article abstractive summarization process. Other types of summarization process are similar. Annotator is in charge of this use case. In the very first step, annotator has to enter the summary that are suitable for the article in the task. When annotator has finished summarizing, annotator will submit the summary by clicking "Gửi kết quả" (Submit) button. Now view will send a HTTP POST Request to server with the request body is summary. In the server side, router will redirect request to corresponding function in SummaryController. This function will call the SummaryService to update the SummaryModel in database through Mongoose. After request has been processed, server will return the response to client side under JSON object. Finally, the result and message will be displayed in view.

b) Use case "Submit review of single article abstractive summarization"

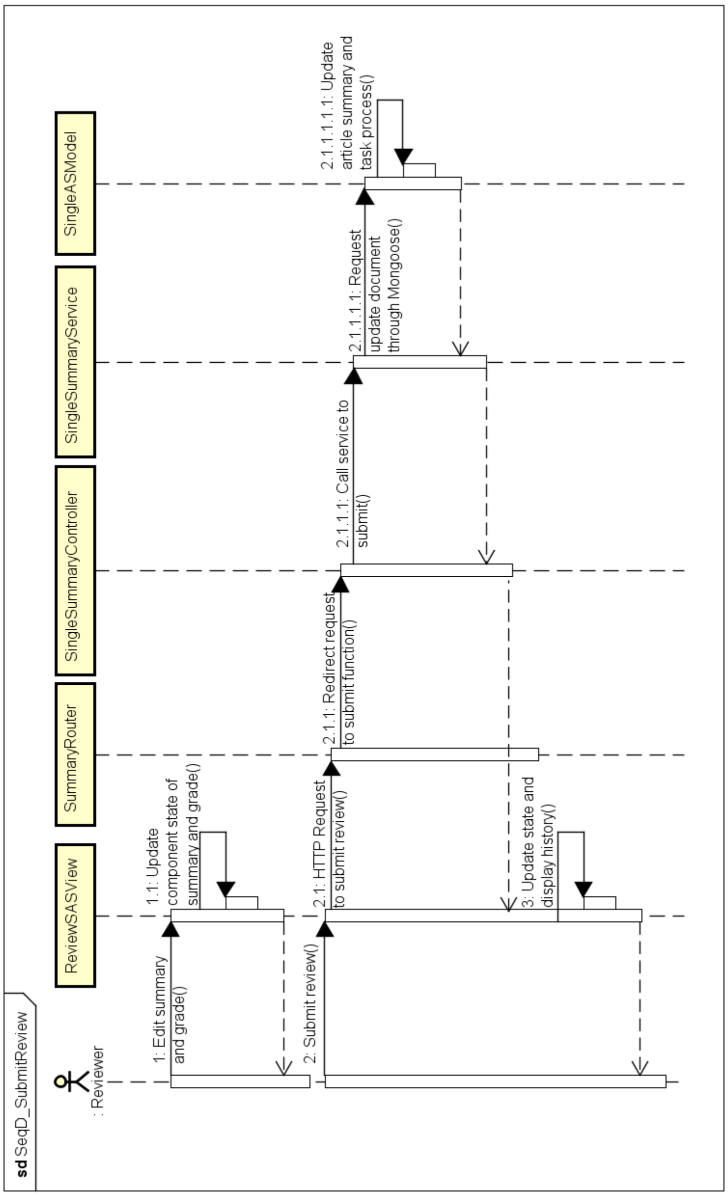
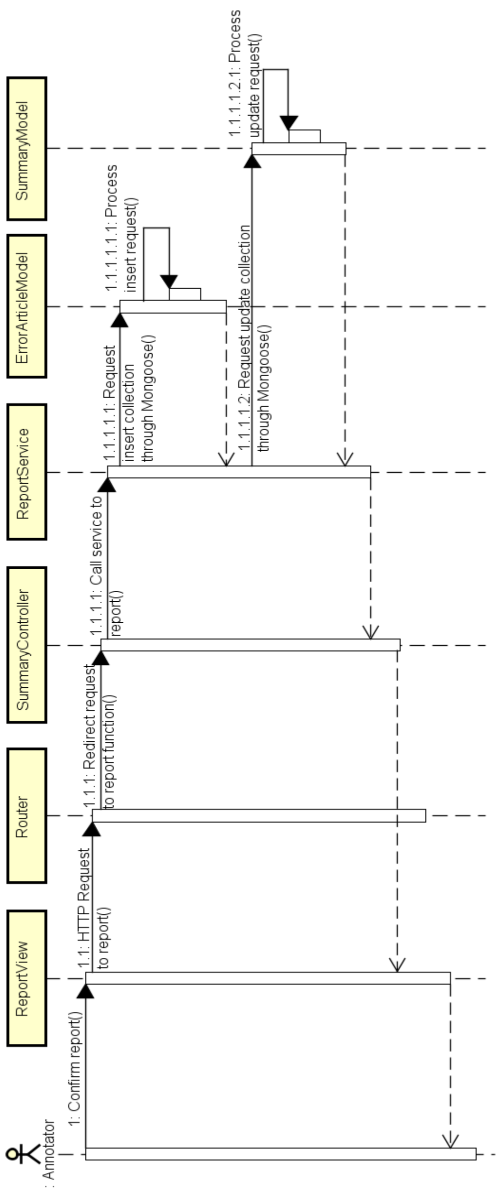

sd SeqD_SubmitReview
: Reviewer
ReviewSASView
SummaryRouter
SingleSummaryController
SingleSummaryService
SingleASModel
1: Edit summary and grade()
1.1: Update component state of summary and grade()
2: Submit review()
2.1: HTTP Request to submit review()
2.1.1: Redirect request to submit function()
2.1.1.1: Call service to submit()
2.1.1.1.1: Request update document through Mongoose()
2.1.1.1.1.1: Update article summary and task process()
3: Update state and display history()

Figure 11: Sequence diagram of "Submit review for single article abstractive summarization".

After summarization process of Annotator, an article will be assigned to Reviewer to be reviewed. In this use case, reviewer can edit the summary which annotator has created and give the score for it. When reviewer submit the review, view will send a HTTP POST Request to server with the body containing summary, comment, and grade. Router in the backend will redirect request to corresponding function in SummaryController. This function will call the SummaryService to update the model in database through Mongoose. After updating the collection in database, server will return the response to client under JSON object. Frontend will display the result.

c) Use case "Report error article"



: Annotator
ReportView
Router
SummaryController
ReportService
ErrorArticleModel
SummaryModel
1: Confirm report()
1.1: HTTP Request to report()
1.1.1: Redirect request to report function()
1.1.1.1: Call service to report()
1.1.1.1.1: Request insert collection through Mongoose()
1.1.1.1.1.1: Process insert request()
1.1.1.1.2: Request update collection through Mongoose()
1.1.1.1.2.1: Process update request()

Figure 12: Sequence diagram of "Report error article".

Any user except Manager can report an article if it is error. However, I would like to present under the annotator perspective only due to the limitation of the thesis. With other users, the process is similar to this.

At first, when user confirm the request, view will make a HTTP POST Request to server. Now the router will redirect the request to the corresponding function in SummaryController. This function will pass the request body to SummaryService to update the status of error article in SummaryModel through Mongoose. After the request is processed successfully, server will return a response to client side and view controller will display the message in view.

4.2.2 General component diagram

4.2.2.1 Frontend component diagram

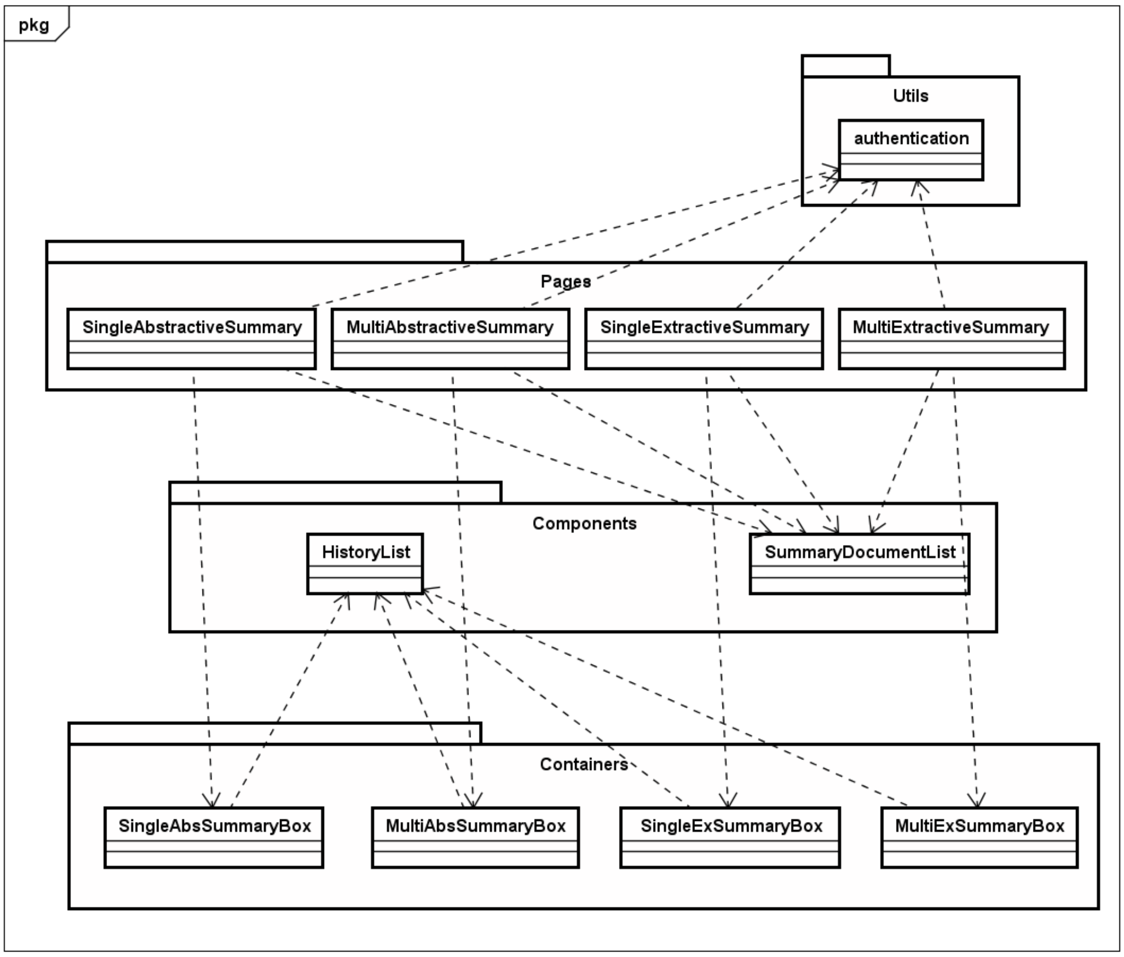


Figure 13: Frontend components.

There are 4 mains packages in frontend: Pages, Containers, Components, and Utils. Package Pages contains 4 pages corresponding to 4 types of text summarization. The Components package has the React components that are reusable in different pages. Meanwhile Containers package consists of the summary box for each type of summarization. These boxes contain other components inside them. All pages depend on "authentication" in Utils package to check for the permission of logged in user.

React is built on components. Each component expresses a certain part of the entire UI of the site, and can be reused in many places without having to copy the source code when you want to use it elsewhere. Container, which can be understood as a bigger component, can have one or many components inside in order to build up a page.

In frontend, components have to connect with the store and follow states. If there are any data that changed, components have to be updated. Moreover, components will process all user behaviors and dispatch to action.

4.2.2.2 Backend component diagram

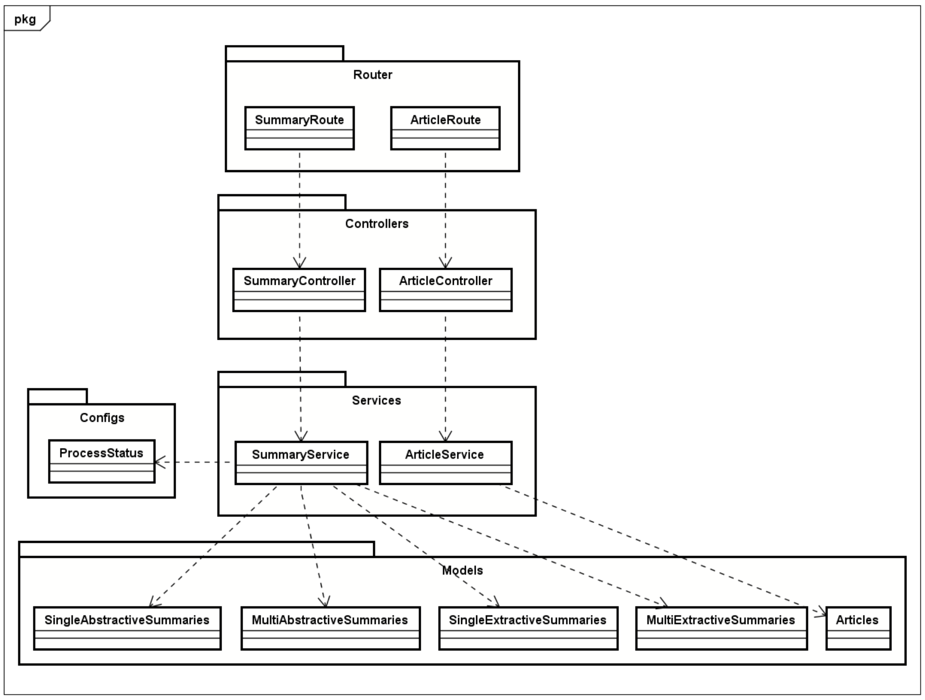


Figure 14: Backend components.

The above figure demonstrates the packages of backend: Router, Controllers, Services, Models, and Configs. Each packages contain the appropriate components that used for summary in my system.

As I mentioned in 4.1.1, Article Service was created and implemented by other student in my group. However, my service needs to get some data about articles' information from Article service, I still have it in the component diagram but I will go into details about my service only, which is Summary Service.

In backend, when receiving HTTP Request from frontend, Summary Router will redirect the request to a function that corresponding to the API in Summary Controller. Then controller will call SummaryService to query the data from Summary Model. There are some functions that needs to call API to get data from Article Service, so those functions in Summary Service have to make HTTP Request to Article Service to retrieve data.

4.3 Detailed design

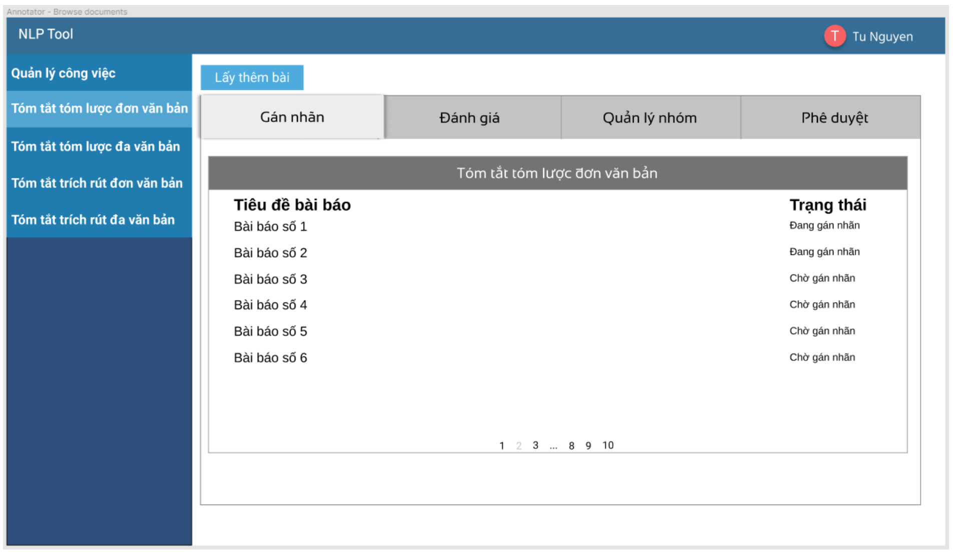
4.3.1 User interface design

The user web interface is designed and built according to React library, so the website will be designed into components. In this part, I would like to introduce 5 interfaces of my webapp. All interfaces have the same layout: (i) Headbar at the top, (ii) Sidebar with navigation menu on the left, and (iii) the rest is for display the necessary data and layout.

The layout of Single type screen and Multiple type screen ("Đơn văn bản", "Đa văn bản") will be similar to each other, therefore, I am going to introduce about the layout of Single abstractive summary and Single extractive summary only.

There are 3 user roles that have the same screen layout: (i) Reviewer, (ii) Group manager, and (iii) Manager. I am going to present the screens for Reviewer only.

4.3.1.1 Task management by role


Annotator - Browse documents
NLP Tool
T Tu Nguyen
Quản lý công việc
Tóm tắt tóm lược đơn văn bản
Tóm tắt tóm lược đa văn bản
Tóm tắt trích rút đơn văn bản
Tóm tắt trích rút đa văn bản
Lấy thêm bài
Gán nhãn
Đánh giá
Quản lý nhóm
Phê duyệt
1 2 3 ... 8 9 10

Tóm tắt tóm lược đơn văn bản

Tiêu đề bài báo	Trạng thái
Bài báo số 1	Đang gán nhãn
Bài báo số 2	Đang gán nhãn
Bài báo số 3	Chờ gán nhãn
Bài báo số 4	Chờ gán nhãn
Bài báo số 5	Chờ gán nhãn
Bài báo số 6	Chờ gán nhãn

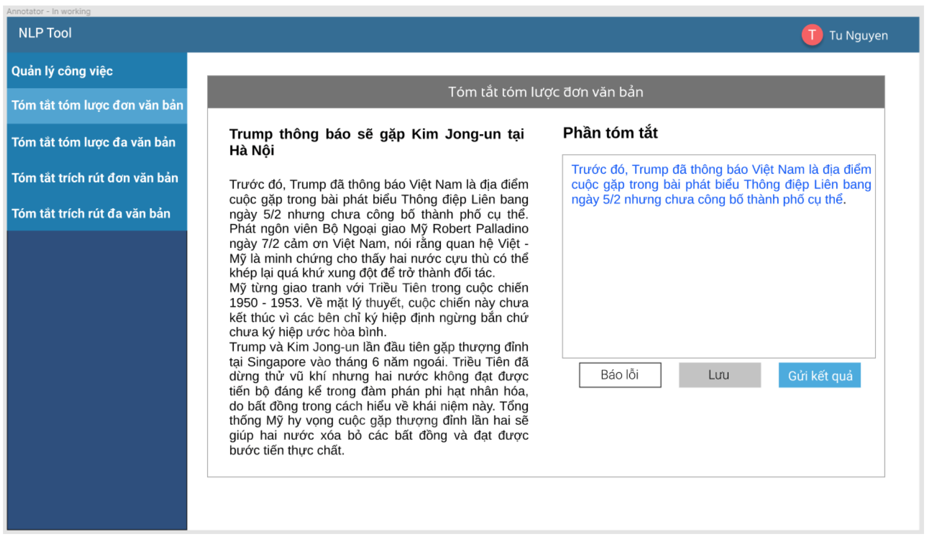
Figure 15: Design list of assigned single abstractive summary tasks of user.

For this screen in Figure 15, each type of summarization contains the same React component, hence I will introduce one type only. The screens of other types are similar to this.

As you can see at the top of the page, under the headbar, there is a button called "Lấy thêm bài" (Request task). Then 4 tabs which are "Gán nhãn", "Đánh giá", "Quản lý nhóm", and "Phê duyệt" ("Annotate", "Review", "Group manage", and "Approve" respectively) is dynamically rendered. These tabs will be shown depending on user roles. The table below each tab is the list of article, or task that user has been assigned.

It has 2 columns: “Tiêu đề bài báo” (Title) and “Trạng thái” (Status). The table also has pagination.

4.3.1.2 Single abstractive summary screen for annotator


Annotator - In working

NLP Tool

T Tu Nguyen

Quản lý công việc

Tóm tắt tóm lược đơn văn bản

Tóm tắt tóm lược đa văn bản

Tóm tắt trích rút đơn văn bản

Tóm tắt trích rút đa văn bản

Tóm tắt tóm lược đơn văn bản

Trump thông báo sẽ gặp Kim Jong-un tại Hà Nội

Trước đó, Trump đã thông báo Việt Nam là địa điểm cuộc gặp trong bài phát biểu Thông điệp Liên bang ngày 5/2 nhưng chưa công bố thành phố cụ thể.
Phát ngôn viên Bộ Ngoại giao Mỹ Robert Palladino ngày 7/2 cảm ơn Việt Nam, nói rằng quan hệ Việt - Mỹ là minh chứng cho thấy hai nước cựu thù có thể khép lại quá khứ xung đột để trở thành đối tác.
Mỹ từng giao tranh với Triều Tiên trong cuộc chiến 1950 - 1953. Về mặt lý thuyết, cuộc chiến này chưa kết thúc vì các bên chỉ ký hiệp định ngừng bắn chứ chưa ký hiệp ước hòa bình.
Trump và Kim Jong-un lần đầu tiên gặp thượng đỉnh tại Singapore vào tháng 6 năm ngoái. Triều Tiên đã dừng thử vũ khí nhưng hai nước không đạt được tiến bộ đáng kể trong đàm phán phi hạt nhân hóa, do bất đồng trong cách hiểu về khái niệm này. Tổng thống Mỹ hy vọng cuộc gặp thượng đỉnh lần hai sẽ giúp hai nước xóa bỏ các bất đồng và đạt được bước tiến thực chất.

Phần tóm tắt

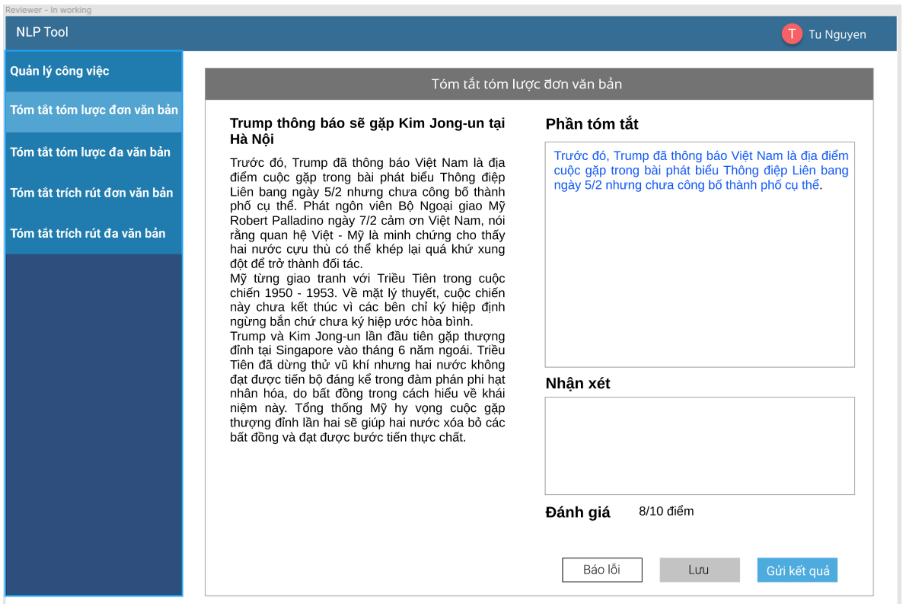
Trước đó, Trump đã thông báo Việt Nam là địa điểm cuộc gặp trong bài phát biểu Thông điệp Liên bang ngày 5/2 nhưng chưa công bố thành phố cụ thể.

Báo lỗi | Lưu | Gửi kết quả

Figure 16: Single abstractive summary task details screen of annotator.

This screen will show details of a task that users have in their list. The layout is quite simple and easy to use. Annotator will read the article on the left of the box, and on the right, there is an input field to type the summary for that article. Below the input field are 3 buttons: “Báo lỗi” (Report error article), “Lưu” (Save), and “Gửi kết quả” (Submit result).

4.3.1.3 Single abstractive summary screen for reviewer


Reviewer - In working

NLP Tool

T Tu Nguyen

Quản lý công việc

Tóm tắt tóm lược đơn văn bản

Tóm tắt tóm lược đa văn bản

Tóm tắt trích rút đơn văn bản

Tóm tắt trích rút đa văn bản

Tóm tắt tóm lược đơn văn bản

Trump thông báo sẽ gặp Kim Jong-un tại Hà Nội

Trước đó, Trump đã thông báo Việt Nam là địa điểm cuộc gặp trong bài phát biểu Thông điệp Liên bang ngày 5/2 nhưng chưa công bố thành phố cụ thể. Phát ngôn viên Bộ Ngoại giao Mỹ Robert Palladino ngày 7/2 cảm ơn Việt Nam, nói rằng quan hệ Việt - Mỹ là minh chứng cho thấy hai nước cựu thù có thể khép lại quá khứ xung đột để trở thành đối tác.
Mỹ từng giao tranh với Triều Tiên trong cuộc chiến 1950 - 1953. Về mặt lý thuyết, cuộc chiến này chưa kết thúc vì các bên chỉ ký hiệp định ngừng bắn chứ chưa ký hiệp ước hòa bình.
Trump và Kim Jong-un lần đầu tiên gặp thượng đỉnh tại Singapore vào tháng 6 năm ngoái. Triều Tiên đã dừng thử vũ khí nhưng hai nước không đạt được tiến bộ đáng kể trong đàm phán phi hạt nhân hóa, do bất đồng trong cách hiểu về khái niệm này. Tổng thống Mỹ hy vọng cuộc gặp thượng đỉnh lần hai sẽ giúp hai nước xóa bỏ các bất đồng và đạt được bước tiến thực chất.

Phần tóm tắt

Trước đó, Trump đã thông báo Việt Nam là địa điểm cuộc gặp trong bài phát biểu Thông điệp Liên bang ngày 5/2 nhưng chưa công bố thành phố cụ thể.

Nhận xét

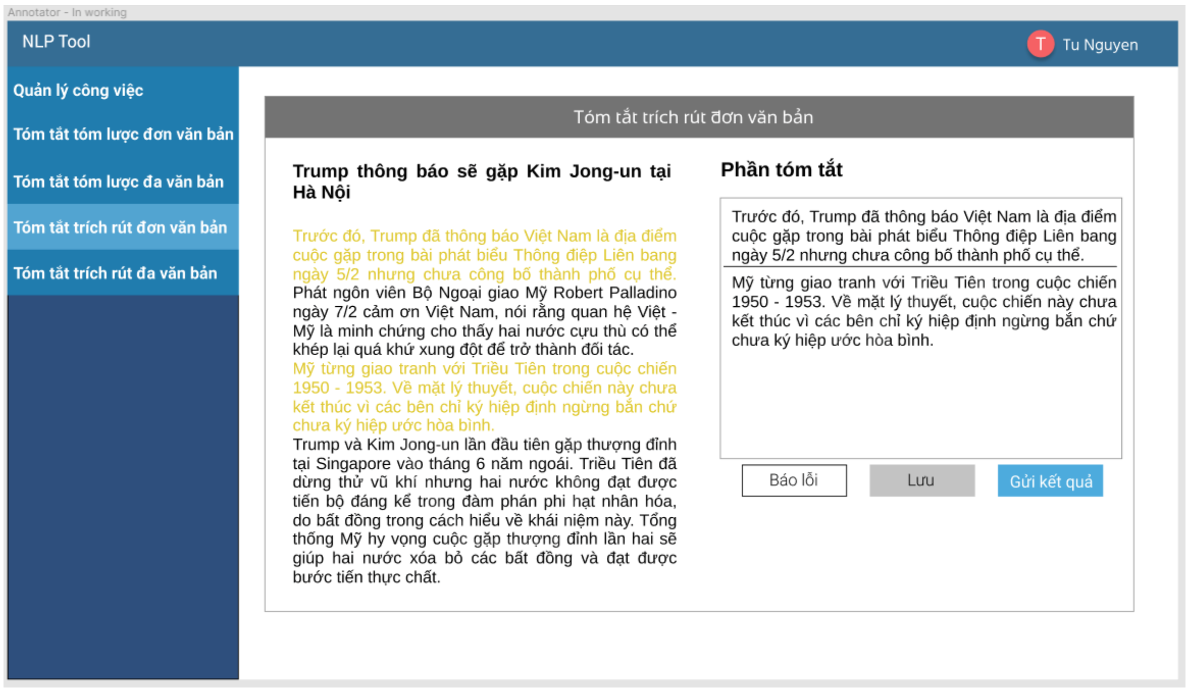
Đánh giá 8/10 điểm

Báo lỗi | Lưu | Gửi kết quả

Figure 17: Single abstractive summary task details screen of reviewer.

When user select task from “Đánh giá” (Review) tab in “Tóm tắt tóm lược đơn văn bản” (Single article abstractive summary), user will be directed to this screen. In general, almost everything is the same with annotator screen except for 2 parts. Reviewer will have the comment input field, which is below “Nhận xét”, and the grade part called “Đánh giá” to give the score for the summary of annotator.

4.3.1.4 Single extractive summary screen for annotator



Annotator - In working
NLP Tool
T Tu Nguyen
Quản lý công việc
Tóm tắt tóm lược đơn văn bản
Tóm tắt tóm lược đa văn bản
Tóm tắt trích rút đơn văn bản
Tóm tắt trích rút đa văn bản
Tóm tắt trích rút đơn văn bản
Báo lỗi
Lưu
Gửi kết quả

Trump thông báo sẽ gặp Kim Jong-un tại Hà Nội

Trước đó, Trump đã thông báo Việt Nam là địa điểm cuộc gặp trong bài phát biểu Thông điệp Liên bang ngày 5/2 nhưng chưa công bố thành phố cụ thể. Phát ngôn viên Bộ Ngoại giao Mỹ Robert Palladino ngày 7/2 cảm ơn Việt Nam, nói rằng quan hệ Việt - Mỹ là minh chứng cho thấy hai nước cựu thù có thể khép lại quá khứ xung đột để trở thành đối tác. Mỹ từng giao tranh với Triều Tiên trong cuộc chiến 1950 - 1953. Về mặt lý thuyết, cuộc chiến này chưa kết thúc vì các bên chỉ ký hiệp định ngừng bắn chứ chưa ký hiệp ước hòa bình. Trump và Kim Jong-un lần đầu tiên gặp thượng đỉnh tại Singapore vào tháng 6 năm ngoái. Triều Tiên đã dừng thử vũ khí nhưng hai nước không đạt được tiến bộ đáng kể trong đàm phán phi hạt nhân hóa, do bất đồng trong cách hiểu về khái niệm này. Tổng thống Mỹ hy vọng cuộc gặp thượng đỉnh lần hai sẽ giúp hai nước xóa bỏ các bất đồng và đạt được bước tiến thực chất.

Phần tóm tắt

Trước đó, Trump đã thông báo Việt Nam là địa điểm cuộc gặp trong bài phát biểu Thông điệp Liên bang ngày 5/2 nhưng chưa công bố thành phố cụ thể.

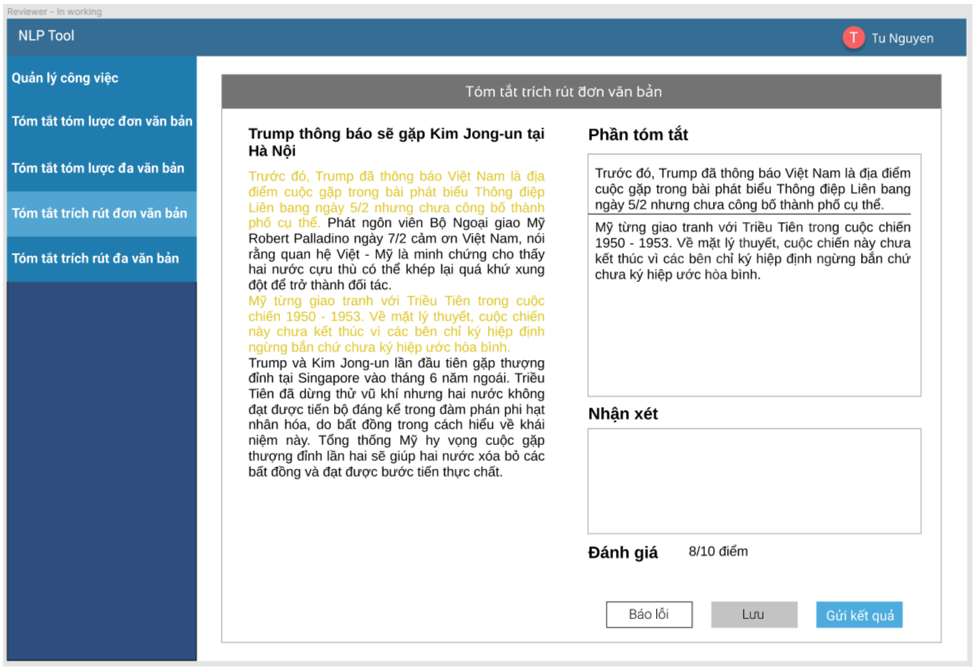
Mỹ từng giao tranh với Triều Tiên trong cuộc chiến 1950 - 1953. Về mặt lý thuyết, cuộc chiến này chưa kết thúc vì các bên chỉ ký hiệp định ngừng bắn chứ chưa ký hiệp ước hòa bình.

Figure 18: Single extractive summary task details screen of annotator.

The figure above show the working screen of annotator in one task of single article extractive summary. In extractive summary, annotator cannot type the summary but choose some sentences from the article’s content and combine them to make a summary. Therefore, color of any sentence in this screen will be changed when it is hover. The input field for summary is now disabled typing, it will contain the selected sentences only. Annotator can click and drag those selected sentences to reorder them. Under the input field is the same 3 buttons as in previous figures.

4.3.1.5 Single extractive summary screen for reviewer

In this screen, almost everything is the same with annotator screen for single extractive summary, except for 2 parts. Reviewer will have the comment input field, which is below “Nhận xét”, and the grade part called “Đánh giá” to give the score for the summary of annotator.


Reviewer - In working
NLP Tool
T Tu Nguyen
Quản lý công việc
Tóm tắt tóm lược đơn văn bản
Tóm tắt tóm lược đa văn bản
Tóm tắt trích rút đơn văn bản
Tóm tắt trích rút đa văn bản
Tóm tắt trích rút đơn văn bản
Báo lỗi
Lưu
Gửi kết quả

Trump thông báo sẽ gặp Kim Jong-un tại Hà Nội

Trước đó, Trump đã thông báo Việt Nam là địa điểm cuộc gặp trong bài phát biểu Thông điệp Liên bang ngày 5/2 nhưng chưa công bố thành phố cụ thể. Phát ngôn viên Bộ Ngoại giao Mỹ Robert Palladino ngày 7/2 cảm ơn Việt Nam, nói rằng quan hệ Việt - Mỹ là minh chứng cho thấy hai nước cựu thù có thể khép lại quá khứ xung đột để trở thành đối tác.
Mỹ từng giao tranh với Triều Tiên trong cuộc chiến 1950 - 1953. Về mặt lý thuyết, cuộc chiến này chưa kết thúc vì các bên chỉ ký hiệp định ngừng bắn chứ chưa ký hiệp ước hòa bình.
Trump và Kim Jong-un lần đầu tiên gặp thượng đỉnh tại Singapore vào tháng 6 năm ngoái. Triều Tiên đã dừng thử vũ khí nhưng hai nước không đạt được tiến bộ đáng kể trong đàm phán phi hạt nhân hóa, do bất đồng trong cách hiểu về khái niệm này. Tổng thống Mỹ hy vọng cuộc gặp thượng đỉnh lần hai sẽ giúp hai nước xóa bỏ các bất đồng và đạt được bước tiến thực chất.

Phần tóm tắt

Trước đó, Trump đã thông báo Việt Nam là địa điểm cuộc gặp trong bài phát biểu Thông điệp Liên bang ngày 5/2 nhưng chưa công bố thành phố cụ thể.

Mỹ từng giao tranh với Triều Tiên trong cuộc chiến 1950 - 1953. Về mặt lý thuyết, cuộc chiến này chưa kết thúc vì các bên chỉ ký hiệp định ngừng bắn chứ chưa ký hiệp ước hòa bình.

Nhận xét

Đánh giá 8/10 điểm

Figure 19: Single extractive summary task details screen of reviewer.

4.3.2 Component design

4.3.2.1 Summary service

This part provides API for request task, get all assigned tasks for user, get article details, save work progress, and submit the result for all types of text summarization. SummaryRoutes will define APIs and redirect the request to appropriate function in SummaryController.

Table 7: API list for single article abstractive summarization

Method	**Endpoint**	**Purpose**
POST	/api/{summaryType}/submit	Submit work result
POST	/api/{summaryType}/save	Save work progress
POST	/api/{summaryType}/requestArticles	Request tasks
POST	/api/suggestSummary	Get suggestion for summary
GET	/api/{summaryType}/{userId}/articles	Get all assigned tasks for user
GET	/api/{summaryType}/articles/{articleId}	Get article details

When calling an API, client has to pass a value for parameter {summaryType}. The valid value for this parameter are: (i) singleAS, (ii) multiAS, (iii) singleES, and (iv) multiES, which are corresponding to Single article abstractive summarization, Multiple article abstractive summarization, Single article extractive summarization, and Multiple article extractive summarization, respectively. Then controller will call functions which will be described in the followings in SummaryService to connect to database.

a) Submit function

This function is call after router redirect from the POST API "/api/{summaryType}/submit" which allows client to submit the work of user to server. The request body is a JSON object with properties: (i) articleId, (ii) userId, (iii) summary, (iv) comment, and (v) rate. ArticleId, userId, and summary are mandatory for all user while reviewer, group manager, and manager must include one more field, which is rate.

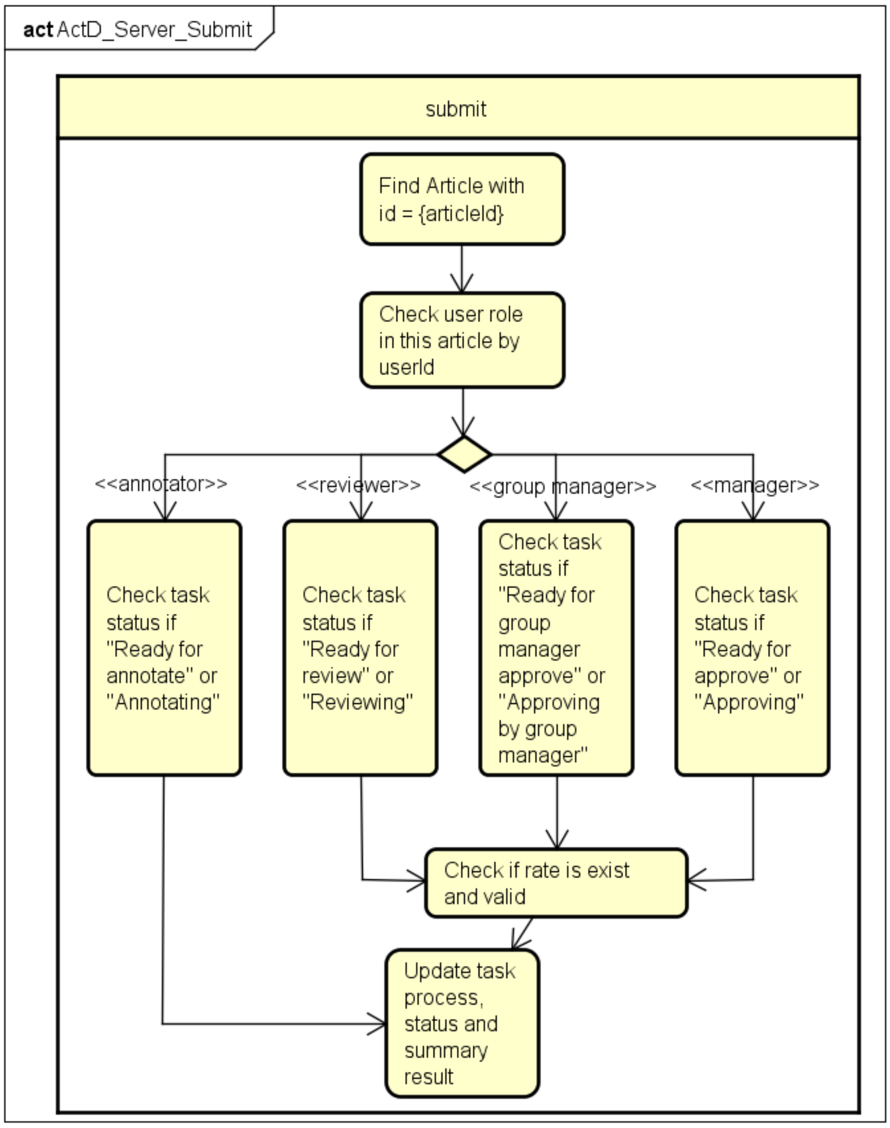

act ActD_Server_Submit
submit
Find Article with id = {articleId}
Check user role in this article by userId
<<annotator>>
<<reviewer>>
<<group manager>>
<<manager>>
Check task status if "Ready for annotate" or "Annotating"
Check task status if "Ready for review" or "Reviewing"
Check task status if "Ready for group manager approve" or "Approving by group manager"
Check task status if "Ready for approve" or "Approving"
Check if rate is exist and valid
Update task process, status and summary result

Figure 20: Activity flow of submit function.

And below is the list of parameters that will be passed into submit function.

Table 8: Parameter list of submit function

Function	**Parameters**	**Mandatory**	**Examples**
submit	articleId	Yes	123
	userId	Yes	5
	summary	Yes	This is a test summary.
	comment	No	This summary is good.
	rate	Yes (except for annotator)	8
	summaryType	Yes	singleAS

Controller will pass to service articleId, userId, and summary. First, service has to check if there exists an article with given articleId. If not, service will return immediately an Error. Then service will check the role of user in this task by comparing the provided userId with annotatorId, reviewerId, groupManagerId, and managerId property in the object. Next, it has to check the current status of the task to make sure it is ready for submit. In addition, except for annotator role, the rest has to check if the rate in request body is valid. (The rate must be an integer number from 1 to 10) Finally, if every field is valid and pass all condition expressions of the function, service will update the document in database. After submission, submit API cannot be called again for this article until its status is changed to an appropriate status.

b) Save function

This function allows client to save the work progress of user to server. The request body is a JSON object with properties: (i) articleId, (ii) userId, (iii) summary, (iv) comment, and (v) rate. ArticleId, userId, and summary are mandatory for all user.

SummaryController will call save function after client make HTTP Request via POST API "/api/{summaryType}/save", which is decribed in the diagram below.

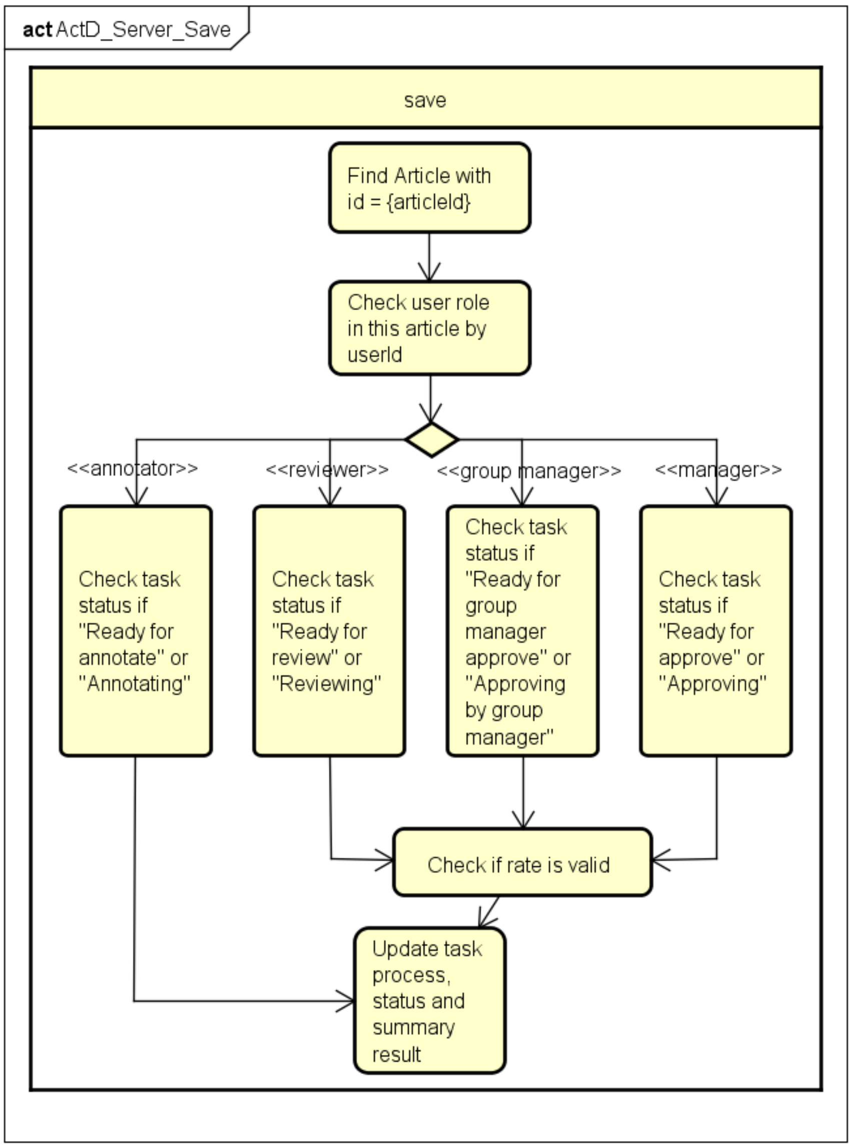

act ActD_Server_Save
save
Find Article with id = {articleId}
Check user role in this article by userId
<<annotator>>
<<reviewer>>
<<group manager>>
<<manager>>
Check task status if "Ready for annotate" or "Annotating"
Check task status if "Ready for review" or "Reviewing"
Check task status if "Ready for group manager approve" or "Approving by group manager"
Check task status if "Ready for approve" or "Approving"
Check if rate is valid
Update task process, status and summary result

Figure 21: Activity flow of save function.

And below tableabove is the list of parameters that will be passed into save function.

Table 9: Parameter list of save function

Function	**Parameters**	**Mandatory**	**Examples**
save	articleId	Yes	123
	userId	Yes	5
	summary	Yes	This is a test summary.
	comment	No	This summary is good.
	rate	No	8
	summaryType	Yes	singleAS

The flow of save function is quite the same as the flow of submit function. The difference between these 2 functions is the status of the task after saving or submitting. Moreover, rate is not required in save function for all user roles. After saving, client can still call save API again or call submit API.

c) RequestArticles function

This function is called from SummaryController after router redirect to when client call POST API: “/api/{summaryType}/requestArticles”. This API allows client to request more task, more article for user, which means users can assign themselves to unassigned tasks. However, user can request if and only if there are 5 or less than 5 tasks remaining in each role. The request body of this API has only 1 property, that is userId.

RequestArticles function is decribed in the diagram below.

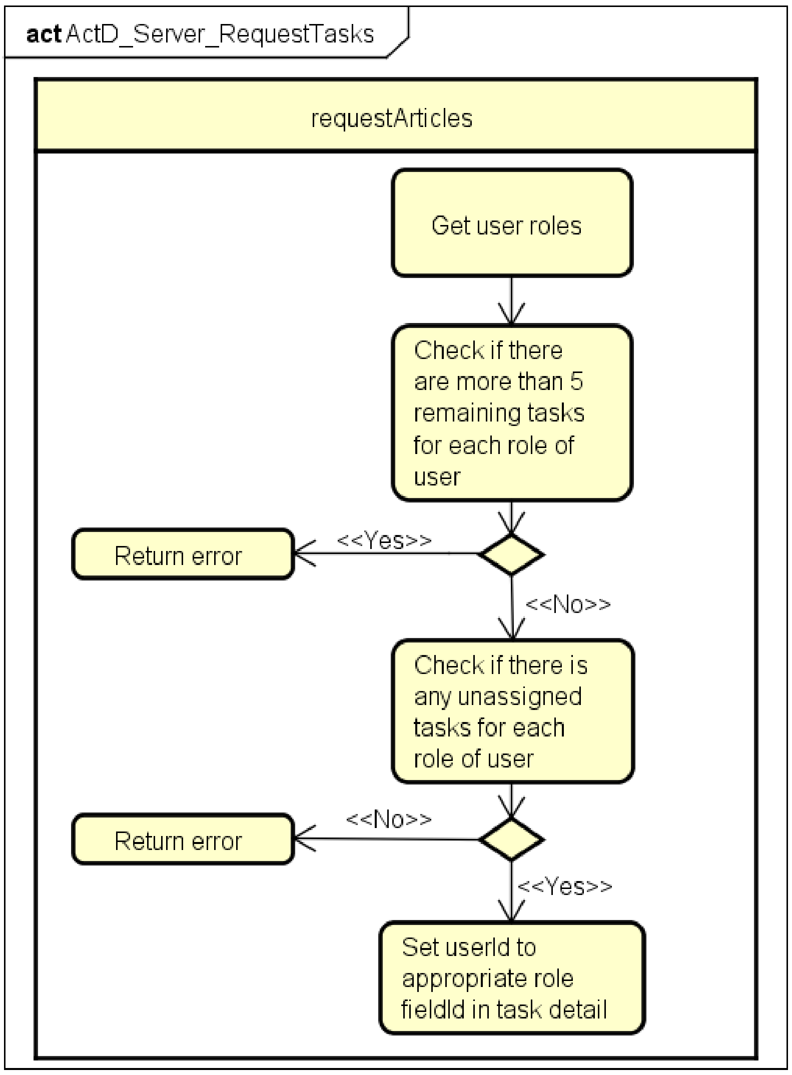

act ActD_Server_RequestTasks
requestArticles
Get user roles
Check if there are more than 5 remaining tasks for each role of user
<<Yes>>
Return error
<<No>>
Check if there is any unassigned tasks for each role of user
<<No>>
Return error
<<Yes>>
Set userId to appropriate role fieldId in task detail

Figure 22: Activity flow of requestArticles function.

Table 10: Parameter list of requestArticles function

Function	Parameters	Mandatory	Examples
requestArticles	userId	Yes	5

Controller will call requestArticles function and pass the userId in request body to this function. At the very first step, it will get the user profiles to know all roles of the user who are making request. Then, for each role, it will check if there are more than 5 remaining tasks. If yes, it will return an error that user cannot request more tasks, or else it will continue to check if there are any unassigned articles, which means free tasks. If exists, service will set the userId to the appropriate role in that task.

d) FindAssignedArticlesByUserId function

GET API: /api/{summaryType}/{userId}/articles will make controller to call this function in service, which allows client to get all current assigned tasks or articles of an user to display to the task list. When calling this API, client has to pass 2 parameters: summaryType and userId. This API will return results containing all assigned articles in all roles of the given user.

Controller will call findAssignedArticlesByUserId() function and pass the summaryType and userId to the function's parameters. The table below shows the list of the function's parameters.

Table 11: Parameter list of findAssignedArticlesByUserId() function

Function	Parameters	Mandatory	Examples
findAssignedArticlesByUserId	userId	Yes	5
	summaryType	Yes	singleAS

The returned results contain article's title and status only.

e) FindArticleById function

This function will be called if client make a HTTP Request to GET API /api/{summaryType}/articles/{articleId}. This allows client to get detail information of an article to display to the working screen of user. This API also has 2 parameters, which is summaryType and articleId.

Controller will call findArticleById() function and pass the summaryType and articleId in query params to function's parameters as shown in the table below. API will return the detail information of the article, which include the annotatorId, reviewerId, groupManagerId, managerId, the process (store the history of each user's work progress), and status.

Table 12: Parameter list of findArticleById() function

Function	Parameters	Mandatory	Examples
findArticleById	articleId	Yes	123
	summaryType	Yes	singleAS

4.3.3 Database design

4.3.3.1 Entity Relation diagram

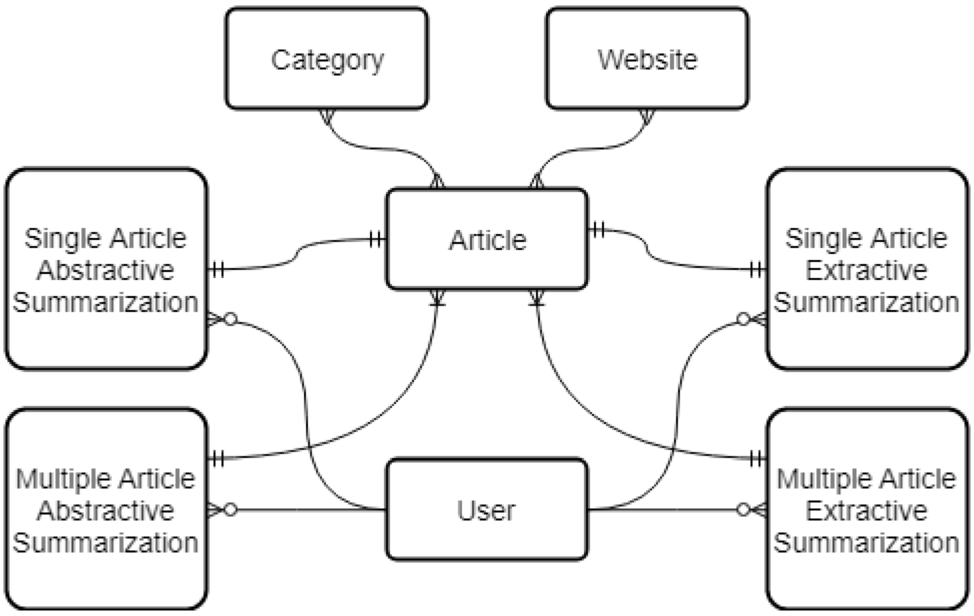

Category
Website
Single Article Abstractive Summarization
Article
Single Article Extractive Summarization
Multiple Article Abstractive Summarization
User
Multiple Article Extractive Summarization

Figure 23: Entity Relation diagram.

It can be seen from Figure 23 that the system has 6 entities. Each article has 1 and only 1 summary from Single article abstractive and extractive summarization. However, for multiple article summarization, each summary of this type can has many articles. An user can summarize as many as possible for any type of summarization.

4.3.3.2 Detail database design

My database has 6 collections, which are article, user, and the rest is for 4 types of summarization. The article collection stores all detail information of articles, which can be status, title, and content. UserId can be the annotatorId, or reviewerId, or groupManagerId, or managerId in each document of each collection of summary. Title and category of article in summary collection are also from the article collection.

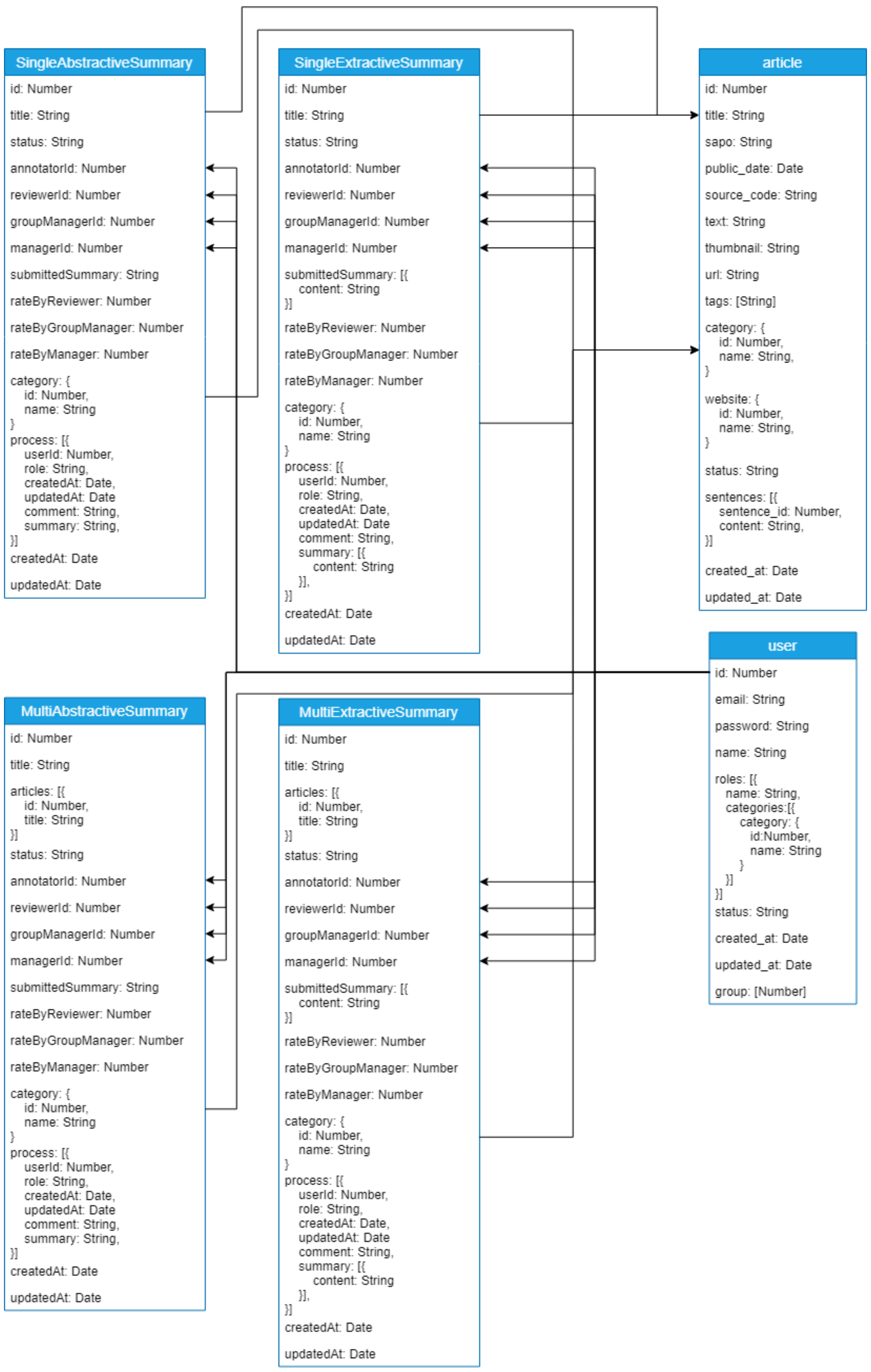


Figure 24: Detail database design.

4.4 Application construction

4.4.1 Library and framework

The following table contains all libraries, frameworks and tools that I used to finish this project.

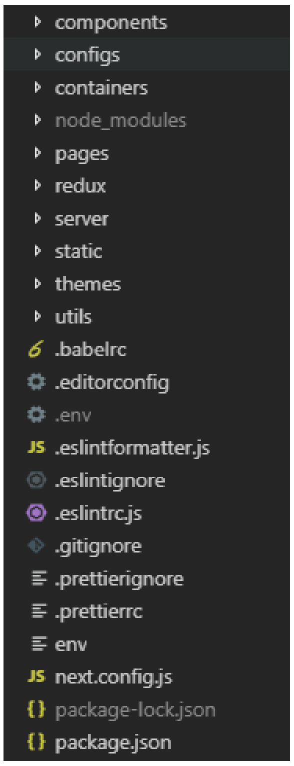
Table 13: Libraries and tools in project

Purpose	**Library / Tool**	**URL**
Code editor	Visual Studio Code	https://code.visualstudio.com/
Package manager	npm	https://www.npmjs.com/
Javascript run-time environment	Nodejs	https://nodejs.org/en/
Javascript library	ReactJs	https://reactjs.org/
React Framework	NextJs	https://nextjs.org/
React UI library	Ant design	https://ant.design/
DBMS	MongoDB	https://www.mongodb.com/
MongoDB Management tool	Robo 3T	https://robomongo.org/
API Development Environment	Postman	https://www.getpostman.com/
UI design tool	Figma	https://www.figma.com/

4.4.2 Results

4.4.2.1 Frontend structure

Web interfaces use the React library to build Redux to manage status. So the structure of the organization includes a server directory which is the server that performs the rendering. In src containing Redux folder, which has actions, store, reducers and React components.

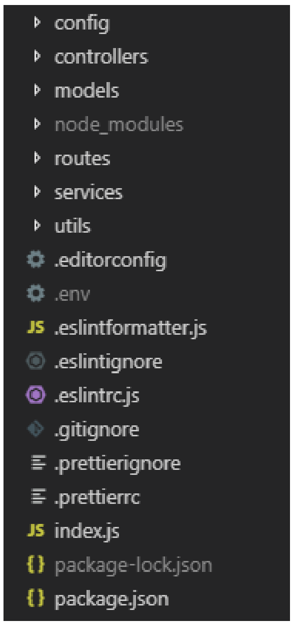


```
components
configs
containers
node_modules
pages
redux
server
static
themes
utils
.babelrc
.editorconfig
.env
.eslintformatter.js
.eslintignore
.eslintrc.js
.gitignore
.prettierignore
.prettierrc
env
next.config.js
package-lock.json
package.json
```

Figure 25: Structure of frontend.

Moreover, because of using NextJs, we have Pages folder which contains all pages of webapp. The Components folder includes components built for common use on multiple screens, enhancing the reusability of components. Configs folder is used to manage the settings for the application.

4.4.2.2 Backend structure



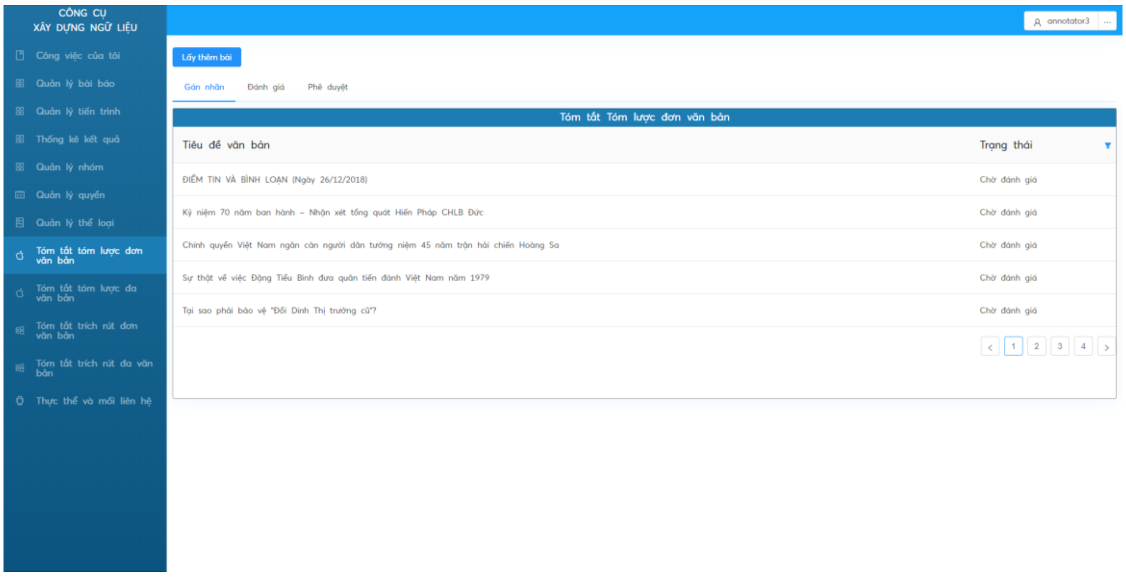
```
config
controllers
models
node_modules
routes
services
utils
.editorconfig
.env
.eslintformatter.js
.eslintignore
.eslintrc.js
.gitignore
.prettierignore
.prettierrc
index.js
package-lock.json
package.json
```

Figure 26: Structure of backend.

Backend in the system has structural elements including: routes containing routes, controllers handling incoming requests and returning response, the controllers are handled through calling to functions in the services; functions in the services call the models to query the database. Config folder manages the configuration and settings of system.

4.4.3 Main screens and features

4.4.3.1 Task view and management screen



Tóm tắt Tóm lược đơn văn bản

Tiêu đề văn bản	Trạng thái
ĐIỂM TIN VÀ BÌNH LOẠN (Ngày 26/12/2018)	Chờ đánh giá
Kỷ niệm 70 năm ban hành – Nhận xét tổng quát Hiến Pháp CHLB Đức	Chờ đánh giá
Chính quyền Việt Nam ngăn cản người dân tưởng niệm 45 năm trận hải chiến Hoàng Sa	Chờ đánh giá
Sự thật về việc Đặng Tiểu Bình đưa quân tiến đánh Việt Nam năm 1979	Chờ đánh giá
Tại sao phải bảo vệ "Đối Dinh Thị trưởng cũ"?	Chờ đánh giá

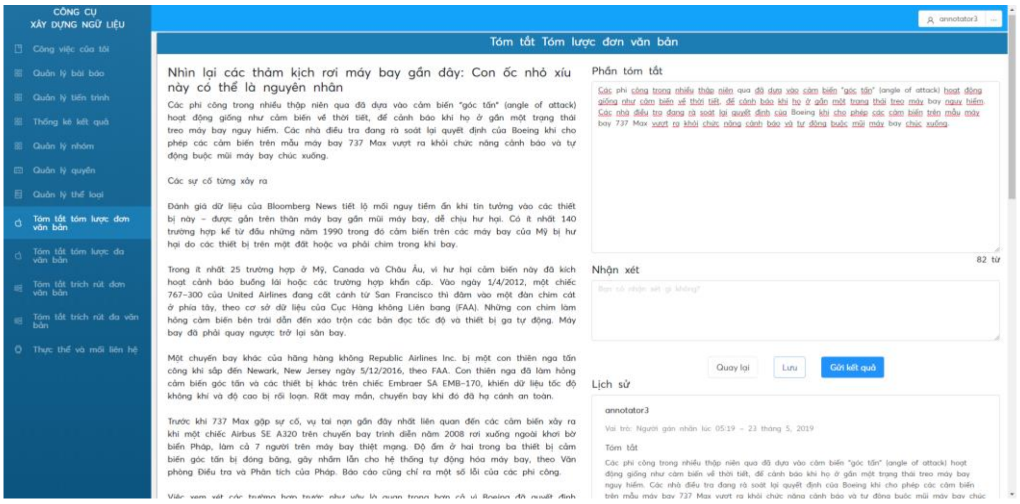
< 1 2 3 4 >

Figure 27: Task list view screen.

This screen displays all the article, or tasks that are assigned to user with specific role in a table with 2 columns: "Tiêu đề văn bản" (Article's title) and "Trạng thái" (Article's status). User can click the icon on the right of "Trạng thái" to filter the displayed article by status. User can select the tab on the top to change the current working role. "Lấy thêm bài" (Request tasks) button is for request more articles if you have finished your work with the old tasks.

4.4.3.2 Annotating screen in Single Abstractive Summarization

This is the screen that Annotator starts working in Single Abstractive Summarization. The below figure is the final product screen for Single abstractive summarization.



Nhìn lại các thảm kịch rơi máy bay gần đây: Con ốc nhỏ xíu này có thể là nguyên nhân

Các phi công trong nhiều thập niên qua đã dựa vào cảm biến "góc tấn" (angle of attack) hoạt động giống như cảm biến về thời tiết, để cảnh báo khi họ ở gần một trạng thái treo máy bay nguy hiểm. Các nhà điều tra đang rà soát lại quyết định của Boeing khi cho phép các cảm biến trên mẫu máy bay 737 Max vượt ra khỏi chức năng cảnh báo và tự động buộc mũi máy bay chúc xuống.

Các sự cố từng xảy ra

Đánh giá dữ liệu của Bloomberg News tiết lộ mối nguy tiềm ẩn khi tin tưởng vào các thiết bị này – được gắn trên thân máy bay gần mũi máy bay, dễ chịu hư hại. Có ít nhất 140 trường hợp kể từ đầu những năm 1990 trong đó cảm biến trên các máy bay của Mỹ bị hư hại do các thiết bị trên mặt đất hoặc va phải chim trong khi bay.

Trong ít nhất 25 trường hợp ở Mỹ, Canada và Châu Âu, vì hư hại cảm biến này đã kích hoạt cảnh báo buồng lái hoặc các trường hợp khẩn cấp. Vào ngày 1/4/2012, một chiếc 767-300 của United Airlines đang cất cánh từ San Francisco thì đâm vào một đàn chim cát ở phía tây, theo cơ sở dữ liệu của Cục Hàng không Liên bang (FAA). Những con chim làm hỏng cảm biến bên trái dẫn đến xáo trộn các bản đọc tốc độ và thiết bị ga tự động. Máy bay đã phải quay ngược trở lại sân bay.

Một chuyến bay khác của hãng hàng không Republic Airlines Inc. bị một con thiên nga tấn công khi sắp đến Newark, New Jersey ngày 5/12/2016, theo FAA. Con thiên nga đã làm hỏng cảm biến góc tấn và các thiết bị khác trên chiếc Embraer SA EMB-170, khiến dữ liệu tốc độ không khí và độ cao bị rối loạn. Rất may mắn, chuyến bay khi đó đã hạ cánh an toàn.

Trước khi 737 Max gặp sự cố, vụ tai nạn gần đây nhất liên quan đến các cảm biến xảy ra khi một chiếc Airbus SE A320 trên chuyến bay trình diễn năm 2008 rơi xuống ngoài khơi bờ biển Pháp, làm cả 7 người trên máy bay thiệt mạng. Độ ẩm ở hai trong ba thiết bị cảm biến góc tấn bị đóng băng, gây nhầm lẫn cho hệ thống tự động hóa máy bay, theo Văn phòng Điều tra và Phân tích của Pháp. Báo cáo cũng chỉ ra một số lỗi của các phi công.

Việc xem xét các trường hợp trước như vậy là quan trọng hơn cả vì Boeing đã quyết định

Phần tóm tắt: Các phi công trong nhiều thập niên qua đã dựa vào cảm biến "góc tấn" (angle of attack) hoạt động giống như cảm biến về thời tiết, để cảnh báo khi họ ở gần một trạng thái treo máy bay nguy hiểm. Các nhà điều tra đang rà soát lại quyết định của Boeing khi cho phép các cảm biến trên mẫu máy bay 737 Max vượt ra khỏi chức năng cảnh báo và tự động buộc mũi máy bay chúc xuống.

annotator3

Vai trò: Người gán nhãn lúc 05:19 - 23 tháng 5, 2019

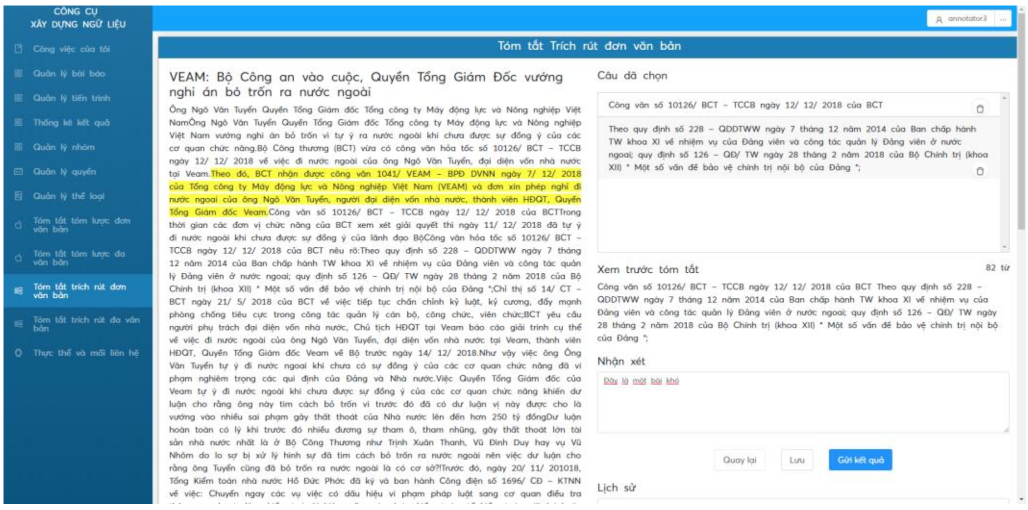
Tóm tắt

Các phi công trong nhiều thập niên qua đã dựa vào cảm biến "góc tấn" (angle of attack) hoạt động giống như cảm biến về thời tiết, để cảnh báo khi họ ở gần một trạng thái treo máy bay nguy hiểm. Các nhà điều tra đang rà soát lại quyết định của Boeing khi cho phép các cảm biến trên mẫu máy bay 737 Max vượt ra khỏi chức năng cảnh báo và tự động buộc mũi máy bay chúc

Figure 28: Single Abstractive Summarization screen with role Annotator.

On the left is the article needed to be summarized. On the other side is the input field where annotator can enter the summary and the comment ("Phần tóm tắt" and "Nhận xét"). The tool always counts word number of the summary and display that at the bottom right of the summary input field. Below are buttons for saving and submitting the summarization result. And all the history of summarization by user are displayed in "Lịch sử" (History) section after the result is saved or submitted.

4.4.3.3 Annotating screen in Single Extractive Summarization



VEAM: Bộ Công an vào cuộc, Quyền Tổng Giám Đốc vướng nghi án bỏ trốn ra nước ngoài

Ông Ngô Văn Tuyến Quyền Tổng Giám đốc Tổng công ty Máy động lực và Nông nghiệp Việt NamÔng Ngô Văn Tuyến Quyền Tổng Giám đốc Tổng công ty Máy động lực và Nông nghiệp Việt Nam vướng nghi án bỏ trốn vì tự ý ra nước ngoài khi chưa được sự đồng ý của các cơ quan chức năng.Bộ Công thương (BCT) vừa có công văn hỏa tốc số 10126/ BCT – TCCB ngày 12/ 12/ 2018 về việc đi nước ngoài của ông Ngô Văn Tuyến, đại diện vốn nhà nước tại Veam.Theo đó, BCT nhận được công văn 1041/ VEAM – BPĐ DVNN ngày 7/ 12/ 2018 của Tổng công ty Máy động lực và Nông nghiệp Việt Nam (VEAM) và đơn xin phép nghỉ đi nước ngoai của ông Ngô Văn Tuyến, người đại diện vốn nhà nước, thành viên HĐQT, Quyền Tổng Giám đốc Veam.Công văn số 10126/ BCT – TCCB ngày 12/ 12/ 2018 của BCTTrong thời gian các đơn vị chức năng của BCT xem xét giải quyết thì ngày 11/ 12/ 2018 đã tự ý đi nước ngoài khi chưa được sự đồng ý của lãnh đạo BộCông văn hỏa tốc số 10126/ BCT – TCCB ngày 12/ 12/ 2018 của BCT nêu rõ:Theo quy định số 228 – QDDTWW ngày 7 tháng 12 năm 2014 của Ban chấp hành TW khoa XI về nhiệm vụ của Đảng viên và công tác quản lý Đảng viên ở nước ngoai; quy định số 126 – QĐ/ TW ngày 28 tháng 2 năm 2018 của Bộ Chính trị (khoa XII) " Một số vấn đề bảo vệ chính trị nội bộ của Đảng ";Chỉ thị số 14/ CT – BCT ngày 21/ 5/ 2018 của BCT về việc tiếp tục chấn chỉnh kỷ luật, kỷ cương, đẩy mạnh phòng chống tiêu cực trong công tác quản lý cán bộ, công chức, viên chức;BCT yêu cầu người phụ trách đại diện vốn nhà nước, Chủ tịch HĐQT tại Veam báo cáo giải trình cụ thể về việc đi nước ngoài của ông Ngô Văn Tuyến, đại diện vốn nhà nước tại Veam, thành viên HĐQT, Quyền Tổng Giám đốc Veam về Bộ trước ngày 14/ 12/ 2018.Như vậy việc ông Ông Văn Tuyến tự ý đi nước ngoai khi chưa có sự đồng ý của các cơ quan chức năng đã vi phạm nghiêm trọng các qui định của Đảng và Nhà nước.Việc Quyền Tổng Giám đốc của Veam tự ý đi nước ngoài khi chưa được sự đồng ý của các cơ quan chức năng khiến dư luận cho rằng ông này tìm cách bỏ trốn vì trước đó đã có dư luận vị này được cho là vướng vào nhiều sai phạm gây thất thoát của Nhà nước lên đến hơn 250 tỷ đồngDư luận hoàn toàn có lý khi trước đó nhiều đương sự tham ô, tham nhũng, gây thất thoát lớn tài sản nhà nước nhất là ở Bộ Công Thương như Trịnh Xuân Thanh, Vũ Đình Duy hay vụ Vũ Nhôm do lo sợ bị xử lý hình sự đã tìm cách bỏ trốn ra nước ngoài nên việc dư luận cho rằng ông Tuyến cũng đã bỏ trốn ra nước ngoài là có cơ sở?!Trước đó, ngày 20/ 11/ 201018, Tổng Kiểm toán nhà nước Hồ Đức Phớc đã ký và ban hành Công điện số 1696/ CĐ – KTNN về việc: Chuyển ngay các vụ việc có dấu hiệu vi phạm pháp luật sang cơ quan điều tra

Câu đã chọn:

Công văn số 10126/ BCT – TCCB ngày 12/ 12/ 2018 của BCT

Theo quy định số 228 – QDDTWW ngày 7 tháng 12 năm 2014 của Ban chấp hành TW khoa XI về nhiệm vụ của Đảng viên và công tác quản lý Đảng viên ở nước ngoai; quy định số 126 – QĐ/ TW ngày 28 tháng 2 năm 2018 của Bộ Chính trị (khoa XII) " Một số vấn đề bảo vệ chính trị nội bộ của Đảng ";

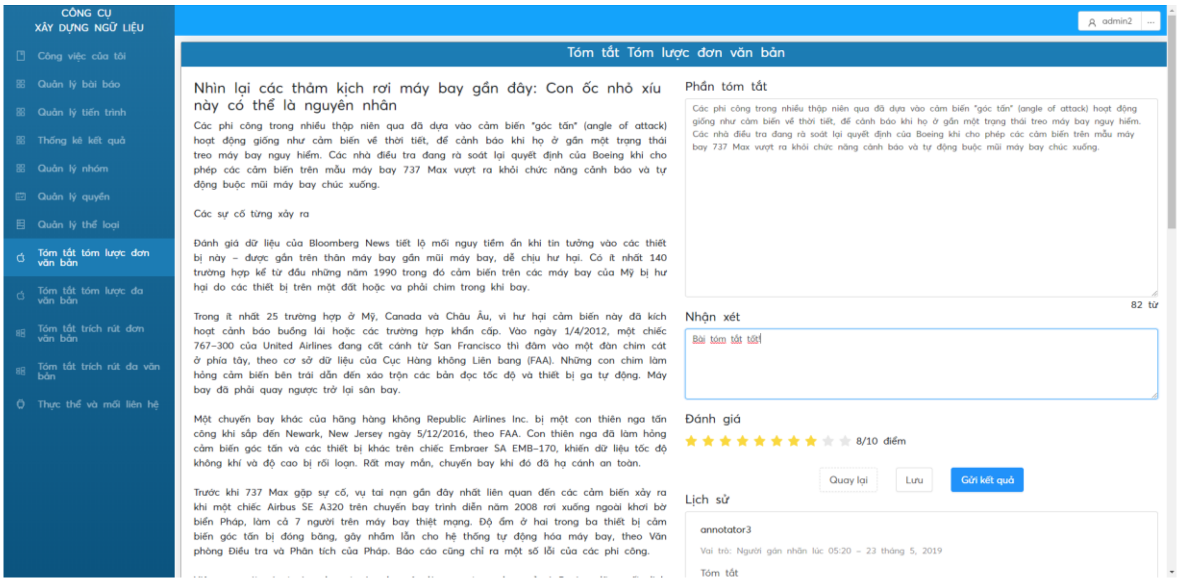
Xem trước tóm tắt: Công văn số 10126/ BCT – TCCB ngày 12/ 12/ 2018 của BCT Theo quy định số 228 – QDDTWW ngày 7 tháng 12 năm 2014 của Ban chấp hành TW khoa XI về nhiệm vụ của Đảng viên và công tác quản lý Đảng viên ở nước ngoai; quy định số 126 – QĐ/ TW ngày 28 tháng 2 năm 2018 của Bộ Chính trị (khoa XII) " Một số vấn đề bảo vệ chính trị nội bộ của Đảng ";

Nhận xét: Đây là một bài khó

Figure 29: Single Extractive Summarization screen with role Annotator.

This is the start working screen of Annotator in Single Extractive Summarization. This one is different from abtractive summarization job is that users cannot enter the summary by themselves. Otherwise, user has to pick up sentences from the original article and combine them to make a summary. User can reorder the selected sentences by drag and drop the sentence in "Câu đã chọn" (Picked sentences) section. At the end of a sentence, there is a delete icon which used for removing the sentence from list of selected sentences. Below is the "Xem trước tóm tắt" (Summary preview) section which show the complete summary from picked sentences. And the rest components are the same with other screens in abstractive summarization.

4.4.3.4 Reviewing screen in Single Abstractive Summarization


CÔNG CỤ XÂY DỰNG NGỮ LIỆU
Công việc của tôi
Quản lý bài báo
Quản lý tiến trình
Thống kê kết quả
Quản lý nhóm
Quản lý quyền
Quản lý thể loại
Tóm tắt tóm lược đơn văn bản
Tóm tắt tóm lược đa văn bản
Tóm tắt trích rút đơn văn bản
Tóm tắt trích rút đa văn bản
Thực thể và mối liên hệ
admin2
Tóm tắt Tóm lược đơn văn bản
Quay lại
Lưu
Gửi kết quả

Nhìn lại các thảm kịch rơi máy bay gần đây: Con ốc nhỏ xíu này có thể là nguyên nhân

Các phi công trong nhiều thập niên qua đã dựa vào cảm biến "góc tấn" (angle of attack) hoạt động giống như cảm biến về thời tiết, để cảnh báo khi họ ở gần một trạng thái treo máy bay nguy hiểm. Các nhà điều tra đang rà soát lại quyết định của Boeing khi cho phép các cảm biến trên mẫu máy bay 737 Max vượt ra khỏi chức năng cảnh báo và tự động buộc mũi máy bay chúc xuống.

Các sự cố từng xảy ra

Đánh giá dữ liệu của Bloomberg News tiết lộ mối nguy tiềm ẩn khi tin tưởng vào các thiết bị này – được gắn trên thân máy bay gần mũi máy bay, dễ chịu hư hại. Có ít nhất 140 trường hợp kể từ đầu những năm 1990 trong đó cảm biến trên các máy bay của Mỹ bị hư hại do các thiết bị trên mặt đất hoặc va phải chim trong khi bay.

Trong ít nhất 25 trường hợp ở Mỹ, Canada và Châu Âu, vì hư hại cảm biến này đã kích hoạt cảnh báo buồng lái hoặc các trường hợp khẩn cấp. Vào ngày 1/4/2012, một chiếc 767–300 của United Airlines đang cất cánh từ San Francisco thì đâm vào một đàn chim cát ở phía tây, theo cơ sở dữ liệu của Cục Hàng không Liên bang (FAA). Những con chim làm hỏng cảm biến bên trái dẫn đến xáo trộn các bản đọc tốc độ và thiết bị ga tự động. Máy bay đã phải quay ngược trở lại sân bay.

Một chuyến bay khác của hãng hàng không Republic Airlines Inc. bị một con thiên nga tấn công khi sắp đến Newark, New Jersey ngày 5/12/2016, theo FAA. Con thiên nga đã làm hỏng cảm biến góc tấn và các thiết bị khác trên chiếc Embraer SA EMB–170, khiến dữ liệu tốc độ không khí và độ cao bị rối loạn. Rất may mắn, chuyến bay khi đó đã hạ cánh an toàn.

Trước khi 737 Max gặp sự cố, vụ tai nạn gần đây nhất liên quan đến các cảm biến xảy ra khi một chiếc Airbus SE A320 trên chuyến bay trình diễn năm 2008 rơi xuống ngoài khơi bờ biển Pháp, làm cả 7 người trên máy bay thiệt mạng. Độ ẩm ở hai trong ba thiết bị cảm biến góc tấn bị đóng băng, gây nhầm lẫn cho hệ thống tự động hóa máy bay, theo Văn phòng Điều tra và Phân tích của Pháp. Báo cáo cũng chỉ ra một số lỗi của các phi công.

Phần tóm tắt

Các phi công trong nhiều thập niên qua đã dựa vào cảm biến "góc tấn" (angle of attack) hoạt động giống như cảm biến về thời tiết, để cảnh báo khi họ ở gần một trạng thái treo máy bay nguy hiểm. Các nhà điều tra đang rà soát lại quyết định của Boeing khi cho phép các cảm biến trên mẫu máy bay 737 Max vượt ra khỏi chức năng cảnh báo và tự động buộc mũi máy bay chúc xuống.

82 từ

Nhận xét

Bài tóm tắt tốt!

Đánh giá

★★★★★★★★☆☆ 8/10 điểm

Lịch sử

annotator3

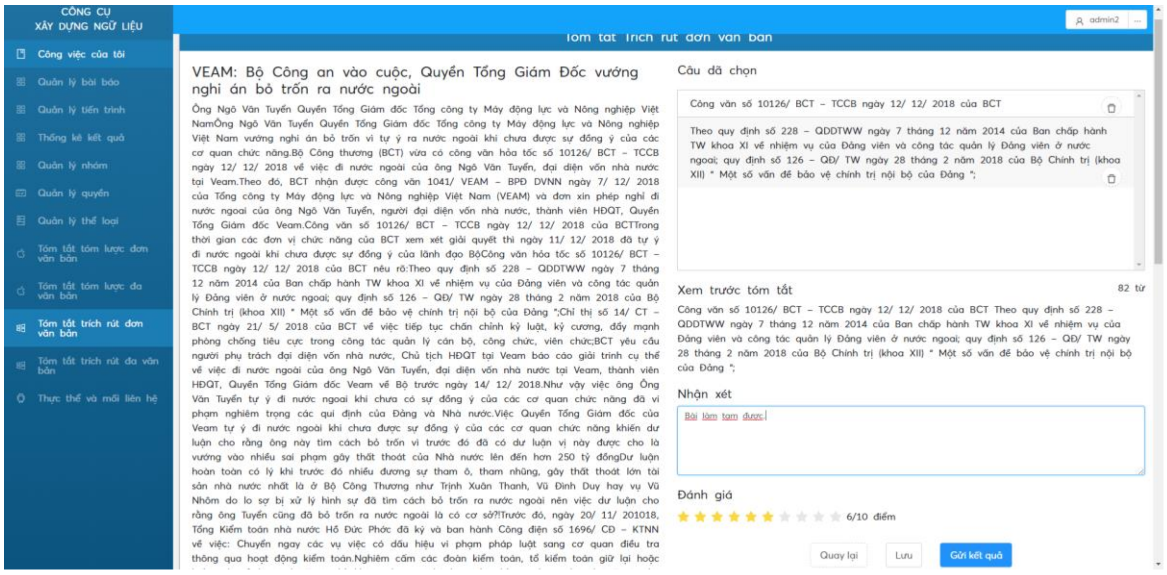
Vai trò: Người gán nhãn lúc 05:20 – 23 tháng 5, 2019

Tóm tắt

Figure 30: Single Abstractive Summarization screen with role Reviewer.

When it comes to reviewing screen, reviewer still can edit and update the summary typed by annotator. However, reviewer is able to grade the summary of annotator by clicking the rate in "Đánh giá" (Rate) section. The grade is in the range of 1 to 10.

4.4.3.5 Reviewing screen in Single Extractive Summarization


CÔNG CỤ XÂY DỰNG NGỮ LIỆU
Công việc của tôi
Quản lý bài báo
Quản lý tiến trình
Thống kê kết quả
Quản lý nhóm
Quản lý quyền
Quản lý thể loại
Tóm tắt tóm lược đơn văn bản
Tóm tắt tóm lược đa văn bản
Tóm tắt trích rút đơn văn bản
Tóm tắt trích rút đa văn bản
Thực thể và mối liên hệ
admin2
Tóm tắt Trích rút đơn văn bản
Quay lại
Lưu
Gửi kết quả

VEAM: Bộ Công an vào cuộc, Quyền Tổng Giám Đốc vướng nghi án bỏ trốn ra nước ngoài

Ông Ngô Văn Tuyển Quyền Tổng Giám đốc Tổng công ty Máy động lực và Nông nghiệp Việt NamÔng Ngô Văn Tuyển Quyền Tổng Giám đốc Tổng công ty Máy động lực và Nông nghiệp Việt Nam vướng nghi án bỏ trốn vì tự ý ra nước ngoài khi chưa được sự đồng ý của các cơ quan chức năng.Bộ Công thương (BCT) vừa có công văn hỏa tốc số 10126/ BCT – TCCB ngày 12/ 12/ 2018 về việc đi nước ngoài của ông Ngô Văn Tuyển, đại diện vốn nhà nước tại Veam.Theo đó, BCT nhận được công văn 1041/ VEAM – BPĐ DVNN ngày 7/ 12/ 2018 của Tổng công ty Máy động lực và Nông nghiệp Việt Nam (VEAM) và đơn xin phép nghỉ đi nước ngoai của ông Ngô Văn Tuyển, người đại diện vốn nhà nước, thành viên HĐQT, Quyền Tổng Giám đốc Veam.Công văn số 10126/ BCT – TCCB ngày 12/ 12/ 2018 của BCTTrong thời gian các đơn vị chức năng của BCT xem xét giải quyết thì ngày 11/ 12/ 2018 đã tự ý đi nước ngoài khi chưa được sự đồng ý của lãnh đạo BộCông văn hỏa tốc số 10126/ BCT – TCCB ngày 12/ 12/ 2018 của BCT nêu rõ:Theo quy định số 228 – QDDTWW ngày 7 tháng 12 năm 2014 của Ban chấp hành TW khoa XI về nhiệm vụ của Đảng viên và công tác quản lý Đảng viên ở nước ngoai; quy định số 126 – QĐ/ TW ngày 28 tháng 2 năm 2018 của Bộ Chính trị (khoa XII) " Một số vấn đề bảo vệ chính trị nội bộ của Đảng ";Chỉ thị số 14/ CT – BCT ngày 21/ 5/ 2018 của BCT về việc tiếp tục chấn chỉnh kỷ luật, kỷ cương, đẩy mạnh phòng chống tiêu cực trong công tác quản lý cán bộ, công chức, viên chức;BCT yêu cầu người phụ trách đại diện vốn nhà nước, Chủ tịch HĐQT tại Veam báo cáo giải trình cụ thể về việc đi nước ngoài của ông Ngô Văn Tuyển, đại diện vốn nhà nước tại Veam, thành viên HĐQT, Quyền Tổng Giám đốc Veam về Bộ trước ngày 14/ 12/ 2018.Như vậy việc ông Ông Văn Tuyển tự ý đi nước ngoai khi chưa có sự đồng ý của các cơ quan chức năng đã vi phạm nghiêm trọng các qui định của Đảng và Nhà nước.Việc Quyền Tổng Giám đốc của Veam tự ý đi nước ngoài khi chưa được sự đồng ý của các cơ quan chức năng khiến dư luận cho rằng ông này tìm cách bỏ trốn vì trước đó đã có dư luận vị này được cho là vướng vào nhiều sai phạm gây thất thoát của Nhà nước lên đến hơn 250 tỷ đồngDư luận hoàn toàn có lý khi trước đó nhiều đương sự tham ô, tham nhũng, gây thất thoát lớn tài sản nhà nước nhất là ở Bộ Công Thương như Trịnh Xuân Thanh, Vũ Đình Duy hay vụ Vũ Nhôm do lo sợ bị xử lý hình sự đã tìm cách bỏ trốn ra nước ngoài nên việc dư luận cho rằng ông Tuyển cũng đã bỏ trốn ra nước ngoài là có cơ sở?!Trước đó, ngày 20/ 11/ 201018, Tổng Kiểm toán nhà nước Hồ Đức Phớc đã ký và ban hành Công điện số 1696/ CĐ – KTNN về việc: Chuyển ngay các vụ việc có dấu hiệu vi phạm pháp luật sang cơ quan điều tra thông qua hoạt động kiểm toán.Nghiêm cấm các đoàn kiểm toán, tổ kiểm toán giữ lại hoặc

Câu đã chọn

Công văn số 10126/ BCT – TCCB ngày 12/ 12/ 2018 của BCT

Theo quy định số 228 – QDDTWW ngày 7 tháng 12 năm 2014 của Ban chấp hành TW khoa XI về nhiệm vụ của Đảng viên và công tác quản lý Đảng viên ở nước ngoai; quy định số 126 – QĐ/ TW ngày 28 tháng 2 năm 2018 của Bộ Chính trị (khoa XII) " Một số vấn đề bảo vệ chính trị nội bộ của Đảng ";

Xem trước tóm tắt

82 từ

Công văn số 10126/ BCT – TCCB ngày 12/ 12/ 2018 của BCT Theo quy định số 228 – QDDTWW ngày 7 tháng 12 năm 2014 của Ban chấp hành TW khoa XI về nhiệm vụ của Đảng viên và công tác quản lý Đảng viên ở nước ngoai; quy định số 126 – QĐ/ TW ngày 28 tháng 2 năm 2018 của Bộ Chính trị (khoa XII) " Một số vấn đề bảo vệ chính trị nội bộ của Đảng ";

Nhận xét

Bài làm tạm được.

Đánh giá

★★★★★★☆☆☆☆ 6/10 điểm

Figure 31: Single Extractive Summarization screen with role Reviewer.

Reviewing screen in Single Extractive Summarization also provides reviewer the ability of updating the summary. All features are still the same with annotator screen but reviewer can rate the summary of annotator by clicking the stars in “Đánh giá” (Rate) section.

4.5 Testing

4.5.1 Test for “Request tasks” feature

Pre-condition: Exists unassigned tasks in database, user currently has no task.

Table 14: Test case of confirm to request tasks

Test case	Steps	Expected result	Test result
Successfully request tasks when clicking confirm	Go to any type of summarization screen	Display task list view	PASS
	Click “Lấy thêm bài” (Request tasks)	Display confirm modal	PASS
	Click “OK” button	Successfully request tasks. System shows tasks in the task list and successful message	PASS

Table 15: Test case of cancel to request tasks

Test case	Steps	Expected result	Test result
Cancel request tasks when clicking cancel	Go to any type of summarization screen	Display task list view	PASS
	Click “Lấy thêm bài” (Request tasks)	Display confirm modal	PASS
	Click “Cancel” button	System does not request tasks. Task list does not have any task, system does not show any message	PASS

4.5.2 Test for “Save” feature

Pre-condition: Exists assigned tasks for user, task status is correct.

Table 16: Test case of “Save” feature

Test case	Steps	Expected result	Test result
Successfully save work progress when	Go to any type of summarization screen	Display task list view	PASS
	Select an article	Direct to detail page and show article’s detail information	PASS

clicking "Lưu" (Save) button	Enter summary and click "Lưu" (Save)	Successfully save work progress. System shows recent work in "Lịch sử" (History) section. System display successful message.	PASS

4.5.3 Test for "Submit" feature

Pre-condition: Exists assigned tasks for user, task status is correct.

Table 17: Test case of "Submit" feature

Test case	**Steps**	**Expected result**	**Test result**
Successfully submit work result when clicking "Gửi kết quả" (Submit) button	Go to any type of summarization screen	Display task list view	PASS
	Select an article	Direct to detail page and show article's detail information	PASS
	Enter summary and click "Gửi kết quả" (Submit)	Display confirm modal	PASS
	Click "OK" button	Successfully submit work result. System shows recent work in "Lịch sử" (History) section. Task status is changed correctly. System display successful message.	PASS

4.6 Deployment

The system was built to bring a real product to real users. Currently, it is delivered to users. The Ministry of Defense used the tool to conduct sample labeling and the Language Institute is using to build training data for the research. These services and interfaces have also been deployed and run on real IP addresses. Details are shown in the table below:

Table 18: Domain and IP of the website

Name	**Domain**	**IP address**
User interface	corpus-tool.vbee.vn	43.239.223.20
API Provider	api-corpus-tool.vbee.vn	43.239.223.22

Chapter 5 Main contributions

After designing and building the application, Chapter 5 will present outstanding solutions and contributions based on the architectures, including: solutions to use the microservices architecture and the proposal of the professional procedure to build and collect a high qualified datasource for text summarization.

5.1 Microservice architecture

Currently, most applications are built as monolithic blocks. At a small and medium level, such applications are easy to write, easy to test and deploy. However, as the number of users, the number of requests increases, the development in a single block results in making the application more cumbersome and complex. If an error arises, it is difficult to determine where the error is, and must conduct testing on millions of lines of source code.

Understanding the disadvantages of the traditional monolithic model, microservice was designed for splitting large applications into interconnected services. Each service will perform a specific set of functions. In each of these services, traditional block models can be applied. Services build APIs to communicate with each other and with third-party applications. Services are separated so that it is able to operate as independently as possible without being dependent or dependent too much on other services in the system. Third-party interfaces or applications that want to use the system's services will not connect directly to each service, but will connect through an intermediate port in the middle. Intermediary portal is responsible for navigation, load, cache, authentication, access control and monitoring.

The advantage of Microservice is to help reduce the complexity and trouble in large systems; easily develop, repair, maintain, upgrade, change, apply new technologies flexibly to each service without changing the system; easily extend new functions when just adding new services and then connecting to intermediate ports.

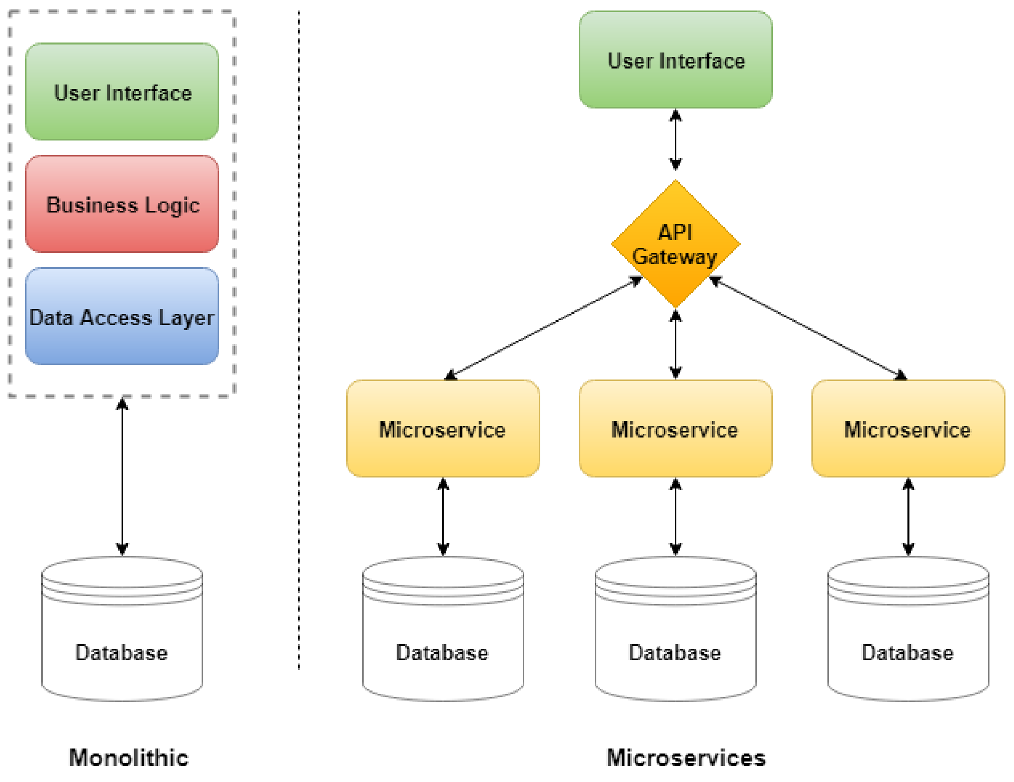


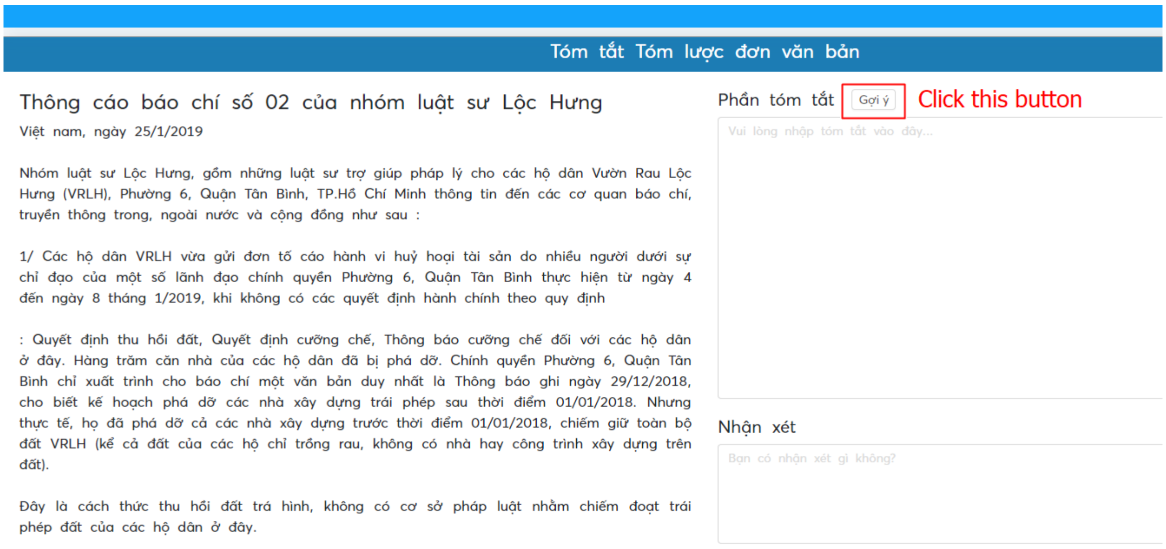
Figure 32: Monolithic vs Microservices.

Start from the requirement of this project, as I have mentioned in Chapter 1, to have a good enough datasource for machine learning in further development, the corpus that we need to build is enormous. Therefore, we need a system, a tool that has high availability and assure to serve a large number of users at the same time. So what I was thinking about was microservice architecture. The entire user interface, administrator interface as well as intermediate portal, and centralized user authentication management services, book management services, authorization management services in the Backend section are designed to run independently on different ports. The Frontend section does not directly connect to the Backend services but connects via an intermediate port. The blocks in the system communicate with each other by calling APIs. These APIs are REST APIs. Under the Microservices architecture model, services will operate independently on separate databases. "Microservice are made to scale large systems. They are great enablers for continuous integration and delivery too."[4]

5.2 Automatic extractive summarization with Text rank

In this tool, I have integrated a suggestion for summary using Text rank algorithm. Details of this function is presented in the following parts.

5.2.1 User interface


Tóm tắt Tóm lược đơn văn bản
Phần tóm tắt
Gợi ý
Click this button
Vui lòng nhập tóm tắt vào đây...
Nhận xét
Bạn có nhận xét gì không?

Thông cáo báo chí số 02 của nhóm luật sư Lộc Hưng

Việt nam, ngày 25/1/2019

Nhóm luật sư Lộc Hưng, gồm những luật sư trợ giúp pháp lý cho các hộ dân Vườn Rau Lộc Hưng (VRLH), Phường 6, Quận Tân Bình, TP.Hồ Chí Minh thông tin đến các cơ quan báo chí, truyền thông trong, ngoài nước và cộng đồng như sau :

1/ Các hộ dân VRLH vừa gửi đơn tố cáo hành vi huỷ hoại tài sản do nhiều người dưới sự chỉ đạo của một số lãnh đạo chính quyền Phường 6, Quận Tân Bình thực hiện từ ngày 4 đến ngày 8 tháng 1/2019, khi không có các quyết định hành chính theo quy định

: Quyết định thu hồi đất, Quyết định cưỡng chế, Thông báo cưỡng chế đối với các hộ dân ở đây. Hàng trăm căn nhà của các hộ dân đã bị phá dỡ. Chính quyền Phường 6, Quận Tân Bình chỉ xuất trình cho báo chí một văn bản duy nhất là Thông báo ghi ngày 29/12/2018, cho biết kế hoạch phá dỡ các nhà xây dựng trái phép sau thời điểm 01/01/2018. Nhưng thực tế, họ đã phá dỡ cả các nhà xây dựng trước thời điểm 01/01/2018, chiếm giữ toàn bộ đất VRLH (kể cả đất của các hộ chỉ trồng rau, không có nhà hay công trình xây dựng trên đất).

Đây là cách thức thu hồi đất trá hình, không có cơ sở pháp luật nhằm chiếm đoạt trái phép đất của các hộ dân ở đây.

Figure 33: Annotator screen with suggest button.

When annotator starts working in abstractive summarization, he/she will see the "Gợi ý" (Suggest) button that is located at top of the screen. If the article is too difficult to be summarized, annotator can click this button to see the suggestion for summary.

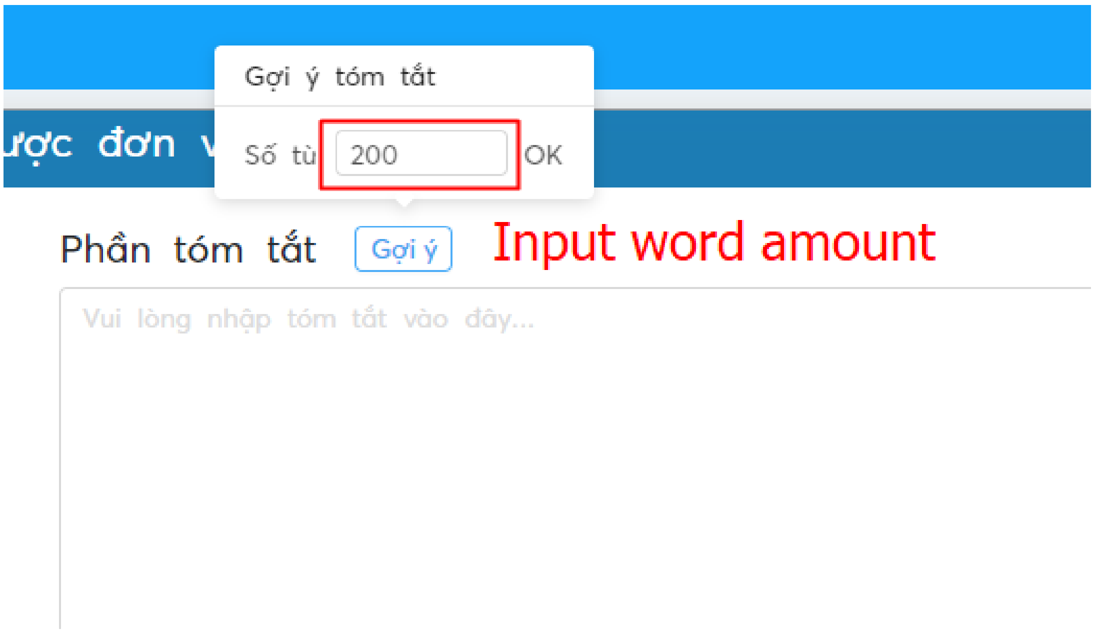
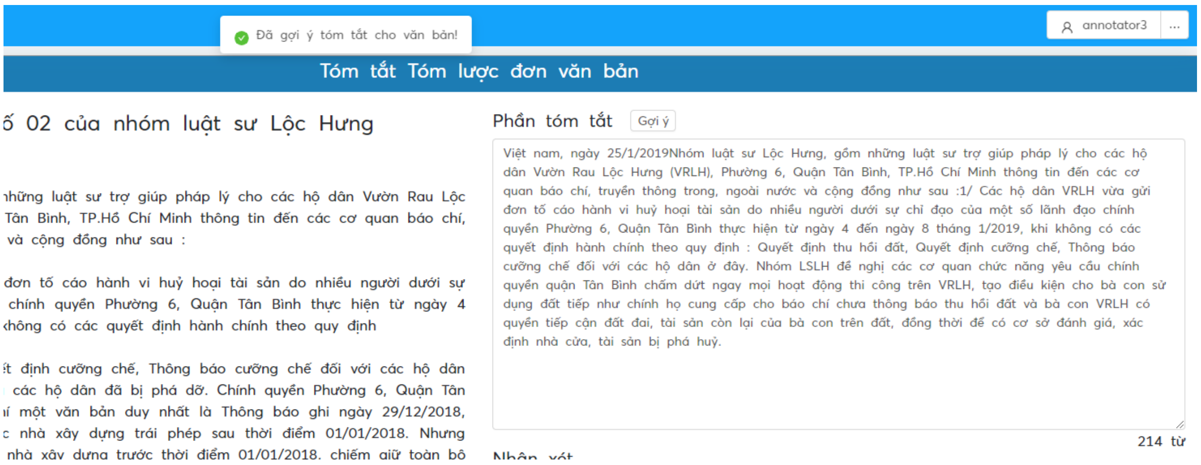


Figure 34: Popover to enter the number of words of summary.

A popover will be displayed when annotator click "Gợi ý" (Suggest) button. There is an input field for user to enter an approximate number of words of the suggest summary. Then annotator click "OK" button to call API to get the result.


Đã gợi ý tóm tắt cho văn bản!

annotator3 ...

Tóm tắt Tóm lược đơn văn bản

ố 02 của nhóm luật sư Lộc Hưng

hững luật sư trợ giúp pháp lý cho các hộ dân Vườn Rau Lộc
Tân Bình, TP.Hồ Chí Minh thông tin đến các cơ quan báo chí,
và cộng đồng như sau :

đơn tố cáo hành vi huỷ hoại tài sản do nhiều người dưới sự
chính quyền Phường 6, Quận Tân Bình thực hiện từ ngày 4
không có các quyết định hành chính theo quy định

ết định cưỡng chế, Thông báo cưỡng chế đối với các hộ dân
các hộ dân đã bị phá dỡ. Chính quyền Phường 6, Quận Tân
í một văn bản duy nhất là Thông báo ghi ngày 29/12/2018,
c nhà xây dựng trái phép sau thời điểm 01/01/2018. Nhưng
nhà xây dựng trước thời điểm 01/01/2018, chiếm giữ toàn bộ

Phần tóm tắt Gợi ý

Việt nam, ngày 25/1/2019Nhóm luật sư Lộc Hưng, gồm những luật sư trợ giúp pháp lý cho các hộ dân Vườn Rau Lộc Hưng (VRLH), Phường 6, Quận Tân Bình, TP.Hồ Chí Minh thông tin đến các cơ quan báo chí, truyền thông trong, ngoài nước và cộng đồng như sau :1/ Các hộ dân VRLH vừa gửi đơn tố cáo hành vi huỷ hoại tài sản do nhiều người dưới sự chỉ đạo của một số lãnh đạo chính quyền Phường 6, Quận Tân Bình thực hiện từ ngày 4 đến ngày 8 tháng 1/2019, khi không có các quyết định hành chính theo quy định : Quyết định thu hồi đất, Quyết định cưỡng chế, Thông báo cưỡng chế đối với các hộ dân ở đây. Nhóm LSLH đề nghị các cơ quan chức năng yêu cầu chính quyền quận Tân Bình chấm dứt ngay mọi hoạt động thi công trên VRLH, tạo điều kiện cho bà con sử dụng đất tiếp như chính họ cung cấp cho báo chí chưa thông báo thu hồi đất và bà con VRLH có quyền tiếp cận đất đai, tài sản còn lại của bà con trên đất, đồng thời để có cơ sở đánh giá, xác định nhà cửa, tài sản bị phá huỷ.

214 từ

Nhận xét

Figure 35: Suggest summary is returned and displayed.

System will return the result and display it inside the summary input field ("Phần tóm tắt" section). As you can see, the suggest summary has 214 words, which is approximately the input of number of words in Figure 34.

5.2.2 Backend and API

When annotator click this button, client will make a HTTP Request to server through POST API /api/suggestSummary to call api "text-summarization.vbee.vn/textrank/".

Table 19: Request body for suggest summary API

Function	**Request body**	**Mandatory**	**Examples**
suggestSummary	document	Yes	Article's content
	summarySize	Yes	200

The above table shows the request body for this API. System will insert the article's content to "document" field and "summarySize" is the number of words that user has input in UI.

With this feature of suggestion for summary, annotators can work easier in the system if they have to face a situation that the article is too difficult to understand and they do not know what to do with the summary. Hence suggest summary will be a supporter for them to finish the work.

5.3 Business process for constructing high quality corpus

As the project is the part of National technology development and application research project, I have been raised the awareness of the importance of this project. The project was born to serve for the automatic crawling, multi-label classification, relation extraction, extractive and abstractive summarization for single and multiple documents of Vietnamese news from Internet, and the purpose is to support and ensure the cyber security. I have proposed the professional procedure to ensure the quality of the output of this web-based tool, which means the corpus for text summarization. In particular, the process can be demonstrated as in the following figure:

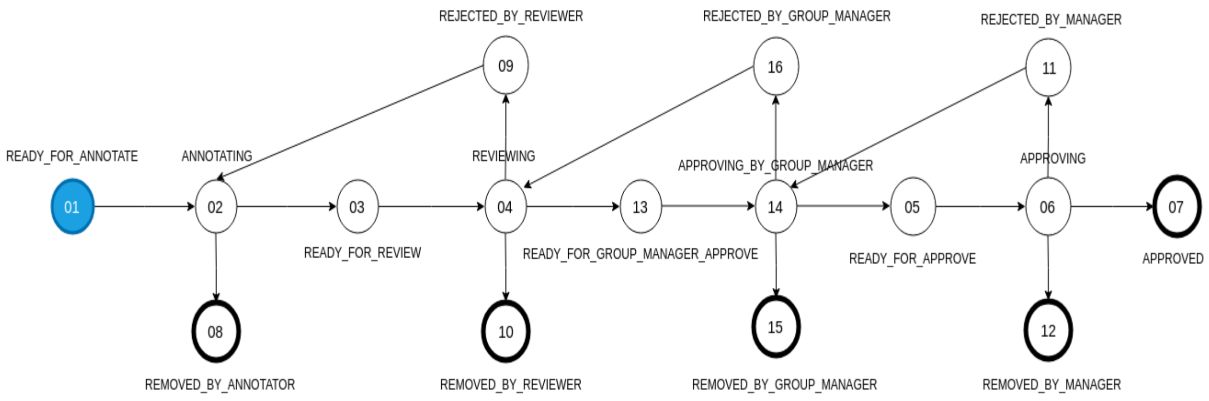


Figure 36: Full business process status of system.

It can be seen from the figure that a summary to be created and approved has to pass many steps with review and approval. When a summary is created by annotator, it must come through 3 steps of reviewing, analyzing, and grading before can be final approved by the manager. With a normal supply chain, double-check is one of the most important part to ensure the product has been qualified enough before reaching to customers. In this project procedure, I can call this is triple-check. The result must be reviewed by reviewer, approved by group manager, and lastly, approved by manager. Moreover, reviewer does not know who created this summary. We hide the information of previous step user so as to ensure that the reviewer in latter step will work in a fair and high quality task. If a task does not meet the need of quality, it will be rejected to make the annotator to redo.

Chapter 6 Conclusion and Future works

Chapter 5 has presented difficulties as well as constructive solutions to solve some complex problems in the application development process. Chapter 6 will summarize the results as well as lessons learned during the application development process in general and graduation project in particular, as well as the orientation of future application development.

6.1 Conclusion

I have presented about the web-based tool for constructing text summarization corpus. The system has been applied with the microservice architecture, using the ReactJs library and NextJs framework for UI building. All services that communicate with each other using the RestfulAPI, which is the simple HTTP protocols that all common programming languages support with NodeJS for the back-end. With the use of API application programming interfaces, the modules in the system are clearly separated, so it is easy to upgrade, repair or replace functions.

The project has reached some good results after a semester of learning new technologies, analyzing the requirement of the system as well as the desire of end user. The tool has the pretty good interface, user friendly and good user experiences. The website is built with web responsive so it is able to be used on different types of monitors. The system has been deployed to product environment, which means it is used by real user, in real conditions now with good feedback.

However, there are some disadvantages that I hope to improve in the near future that is the business process is a little complicated, which makes user hard to trace the process of a task.

6.2 Future works

In the near future, I would like to improve the process in an easier way but higher quality. Furthermore, I want to improve the UI and user experience in order to have a good environment for user to building better corpus.

Referrences

[1] B. Burkholder, “JavaScript SEO: Server Side Rendering vs Client Side Rendering,” *Medium*, 05-Jul-2018.

[2] ZEIT, Inc., “Learn | Next.js.” [Online]. Available: https://nextjs.org/learn. [Accessed: 23-May-2019].

[3] “What is REST – Learn to create timeless RESTful APIs.” [Online]. Available: https://restfulapi.net/. [Accessed: 21-May-2019].

[4] T. A. Gamage, “Microservices Design Guide,” *Platform Engineer*, 21-Oct-2018.