

UNIVERSITY OF SCIENCE AND TECHNOLOGY OF HANOI
UNDERGRADUATE SCHOOL



FINAL REPORT
BACHELOR GRADUATION

By
Bui Quoc Trung

Project:
MOBILE APPLICATIONS FOR ACQUIRING VIETNAMESE VOICE DATA

Supervisor: Dr. Tran Giang Son
ICTLab, USTH

Hanoi, September 2019

Contents

Contents	2
Acknowledgements	4
List of abbreviations	5
List of Figures	6
I. Introduction	8
1.1 Context and motivation	8
1.2 Objective	8
1.3 Thesis structure	9
II. Material and methods	9
2.1: Data type	9
2.2 Function description	9
2.3 System architecture	10
2.4 Diagrams	11
2.4.1 Use case diagram	11
2.4.2 Class diagrams:	12
2.4.3 Sequense diagram:	13
2.5 Tools and implementation:	17
2.5.1 Client-Server Model	17
2.5.3 Client-side	18
III. Result and Disscusion	18
3.1 Result	18
3.2 Discussion	19
3.2.1 Why flutter?	19

3.2.2 Difficult	19
VI. Conclusion and Future works	20

Acknowledgements

I would like to express my special thanks of gratitude to my teacher Dr. TranGian Son who gave me the opportunity to do this wonderful project on the topic Mobile applications for acquiring Vietnamese voice data, which also helped me inenriching experience and i came to know about so many new things.

List of abbreviations

List of Figures

1	System architecture diagram	10
2	Use case diagram	11
3	Class diagram	12
4	Sequense diagram for Logging in funtion	13
5	Sequense diagram for Recording and Uploading function	14
6	Sequense diagram for Voting function	15
7	Sequense diagram for Editting fucntion	16
8	Client-Server Model	17

Abstracts

Today many developers, researchers and startups want to test and build technologies that support natural speech. Although the speech data set is an essential component for building natural language processing tools, the current amount of Vietnamese speech data is quite limited. Voice Viet is a database of Vietnamese voices shared for the community, contributed by state agencies, organizations, businesses and every citizen. Anyone can access the Voice Vietnamese database to address their needs, such as for speech recognition and speech synthesis applications. To support people can easily access and contribute to the database, I decided to develop a mobile application on Android and IOS platform. The purpose of this paper is to describe my own record application implementation. My applications are written in Java for Android platform and Flutter for IOS platform.

Keywords: Mobile Application, Android, IOS, Java, Flutter

I. Introduction

1.1 Context and motivation

Project “Phát triển Hệ tri thức Việt số hóa” (itrithuc.vn) is assigned by Vietnam Government to Ministry of Science and Technology (MOST). The overall goal of the project is to disseminate knowledge in all fields, first of all supporting education and training, innovation and fields directly related to the lives of people such as law and medicine, production techniques. Creating a favorable environment to attract people and businesses to participate as both an exploiter and contributor to enrich Vietnam’s digital knowledge resources. The project creates an interdisciplinary open data platform with many different types of data (address, map, travel, voice, agriculture, medicine, etc.) Shared data platform Itrithuc is a national knowledge resource in which all organizations and businesses of scientists at home and abroad as well as people can participate and in the process of collecting, sharing and exploiting data on the ground this platform.

Voice Viet is one of the most important parts of Itrithuc platform, which is the database of Vietnamese voice. As the same goal with the Itrithuc platform, this database is shared for everyone can access to contribute and use the Vietnamese voice data.

In this project, I target my motivation on the mobile device and will develop a mobile phone application. Since the smartphone is a very common and inseparable tool among people all around the world in general and Vietnam in particular, developing an application on the mobile platform is extremely suitable. As a result, Vietnamese people can contribute to the database anytime and anywhere with an Internet connection. So it becomes the easiest way for people to approach to the database and make it bigger and more diverse. In the internship period, I tried to develop this application on two biggest platforms, Android and IOS, that accounting for 87.8 percent of the global mobile phone market share.

1.2 Objective

Successfully develop two application on Android and IOS platform with fully function.

- Users can record voice and upload there record to server.
- Users can vote for the voice record.
- Users can edit the text attached with the voice record and replace it on server.
- All voice record can be played.

1.3 Thesis structure

The rest of the thesis will be structured as following:

- **Section 2:** Introduction about data type and architecture including tools and implementation.
- **Section 3:** Present result and discussion.
- **Section 4:** Summarize the result and talk about future improvement

II. Material and methods

2.1: Data type

In my project, the main material used is voice data, specifically Vietnamese voice. Speech data in general is important for speech-to-text and text-to-speech systems. The larger the data storage, the greater the accuracy of the above systems. There are no standards for the data to be collected, however, the audio recording needs to match the accompanying text. The data collected is purely raw, untreated data. Data collectors, age is not limited. The recording environment may have noise but make sure the recording is still audible. The audio recording is limited to 30 seconds in length.

2.2 Function description

In the application the users not only contribute their voice for the database but also they can participate in assessing and editing data. Following is some main functions of the application:

- **Recording and Uploading function:** With this function, users can record their voice by reading an existing text paragraph. User can review their record before uploading to the server. However, the users need to log in before uploading to the server.
- **Voting function:** As I mentioned before the users not only contribute their voice, now they can participate in vote for the records of other users or records are made by machine. By this way, the users can bring up to the accuracy of the data set.
- **Editing function:** In this function, the users will evaluate the text that comes with the audio recording. Because, many text paragraph are made automatically. Now, if the text is not good enough, the users can edit and push them back to server.

2.3 System architecture

The architecture system of the software consists of 3 layers, the top is the Presentation class, in the middle is the processing layer, the last is the data layer. Presentation layer is a place for users to interact with software, communicate and display user requests and then bring requests to the processing layer. The Processing layer will classify and process user requests such as checking single words or translating paragraphs; determine the input data whether they are sound or text to retrieving data from the offline database or use the last layer where storage the application data.

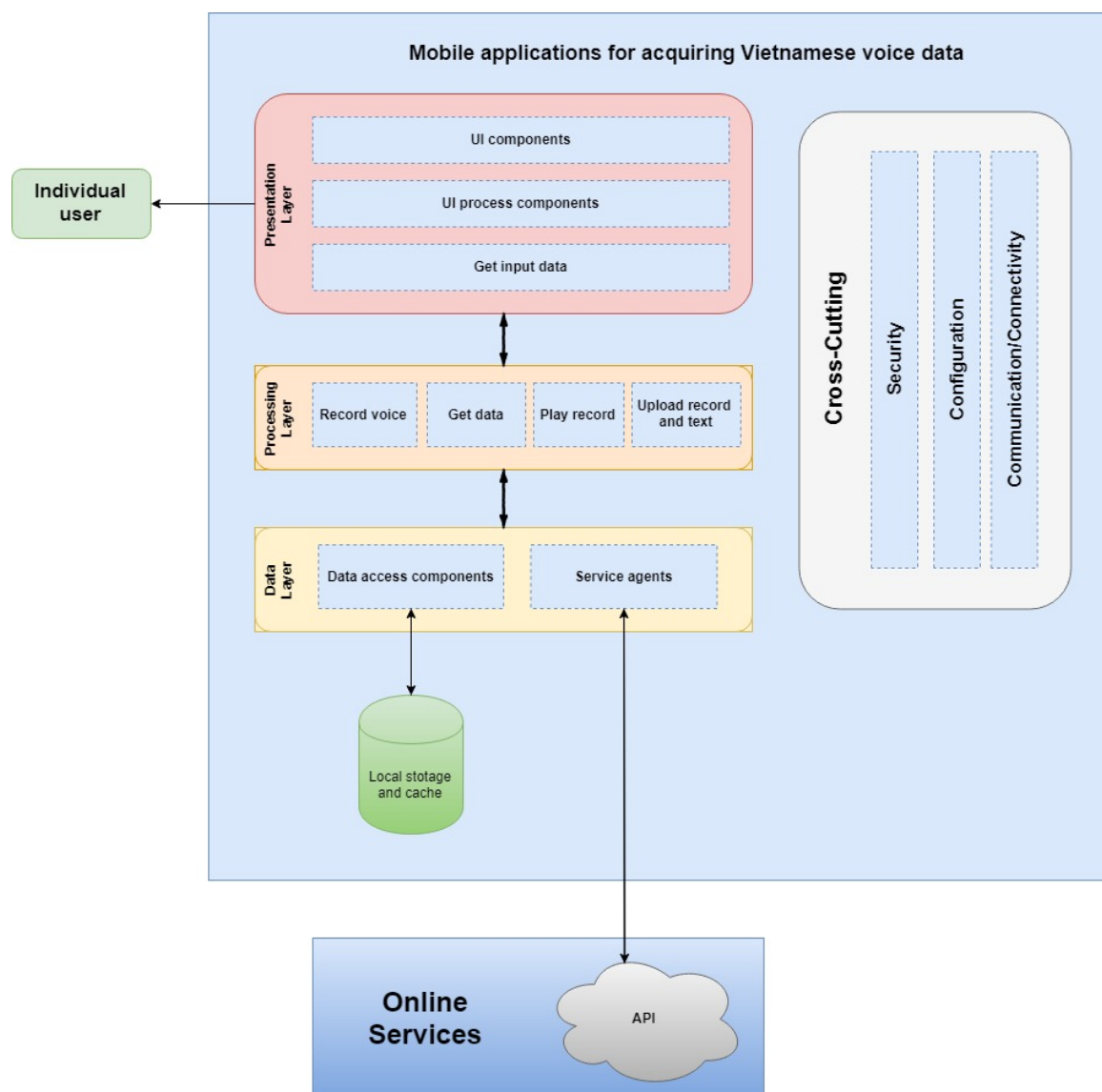


Figure 1: System architecture diagram

2.4 Diagrams

To make it easier to visualize and understand the system of application, here are diagrams to show the different aspects of the application.

2.4.1 Use case diagram

Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform. In this context, a “system” is application, such as a web site. The “actors” are users.

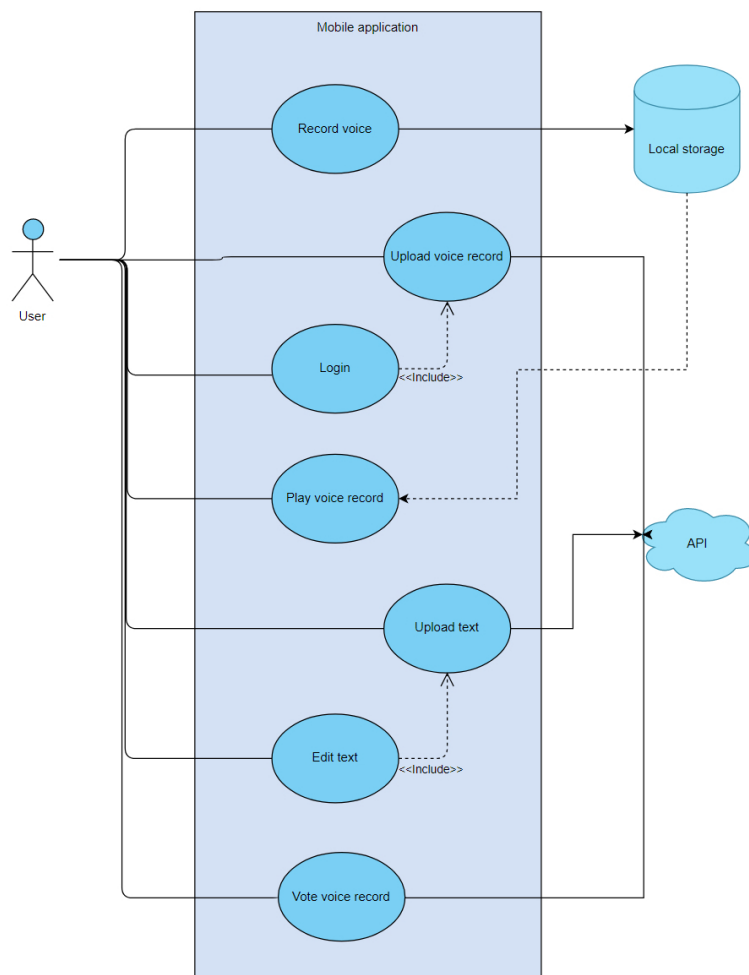


Figure 2: Use case diagram

2.4.2 Class diagrams:

A class diagram models the static structure of a system. It shows relationships between classes, objects, attributes, and operations.

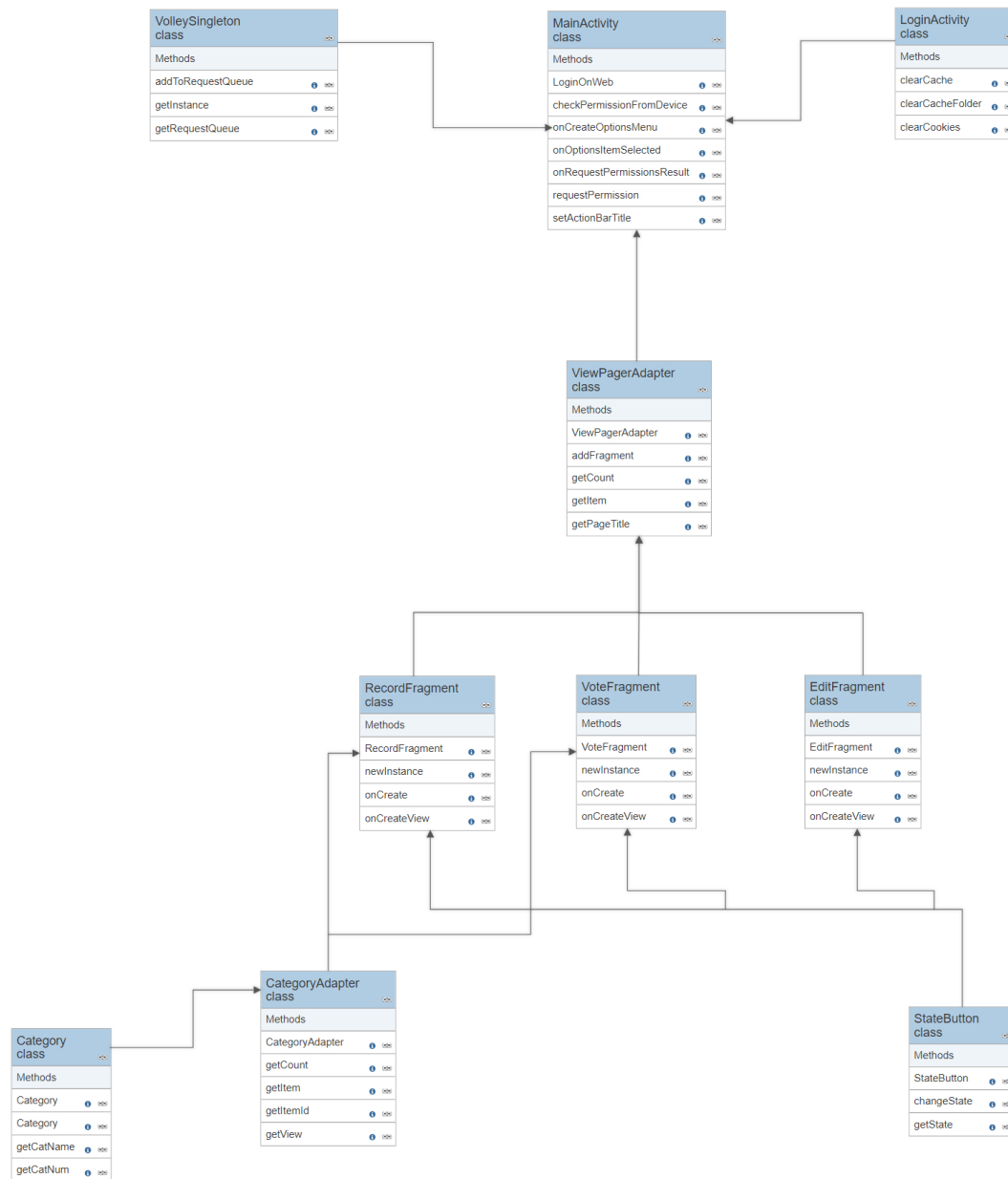


Figure 3: Class diagram

2.4.3 Sequence diagram:

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. Those diagrams following will describe the behavior of system:

- **Figure 4:** Describe the login process of the user.

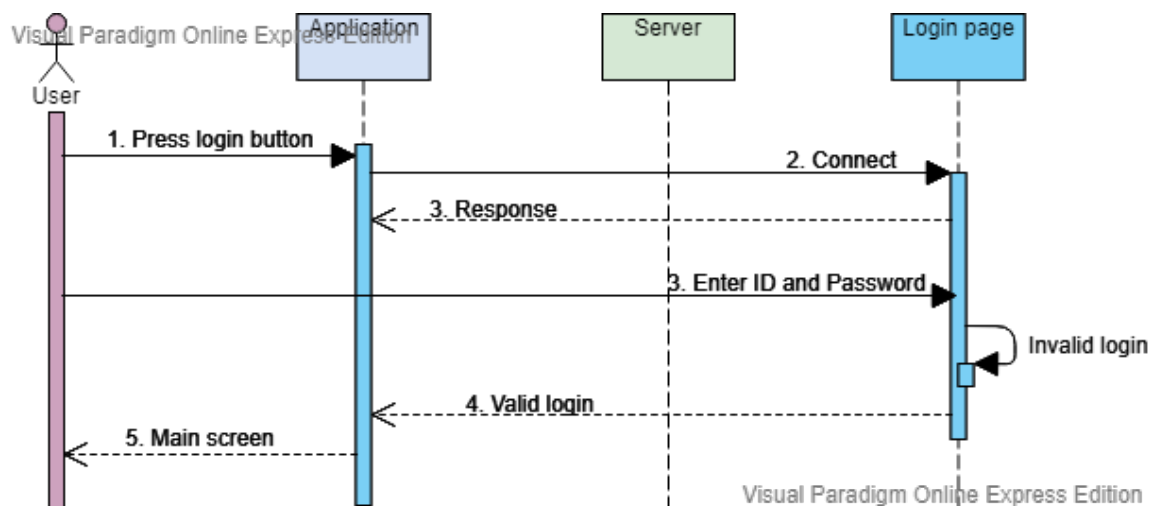


Figure 4: Sequence diagram for Logging in function

- **Figure 5:** Describes the activities that occur on the screen of the recording function and upload the sound recording.

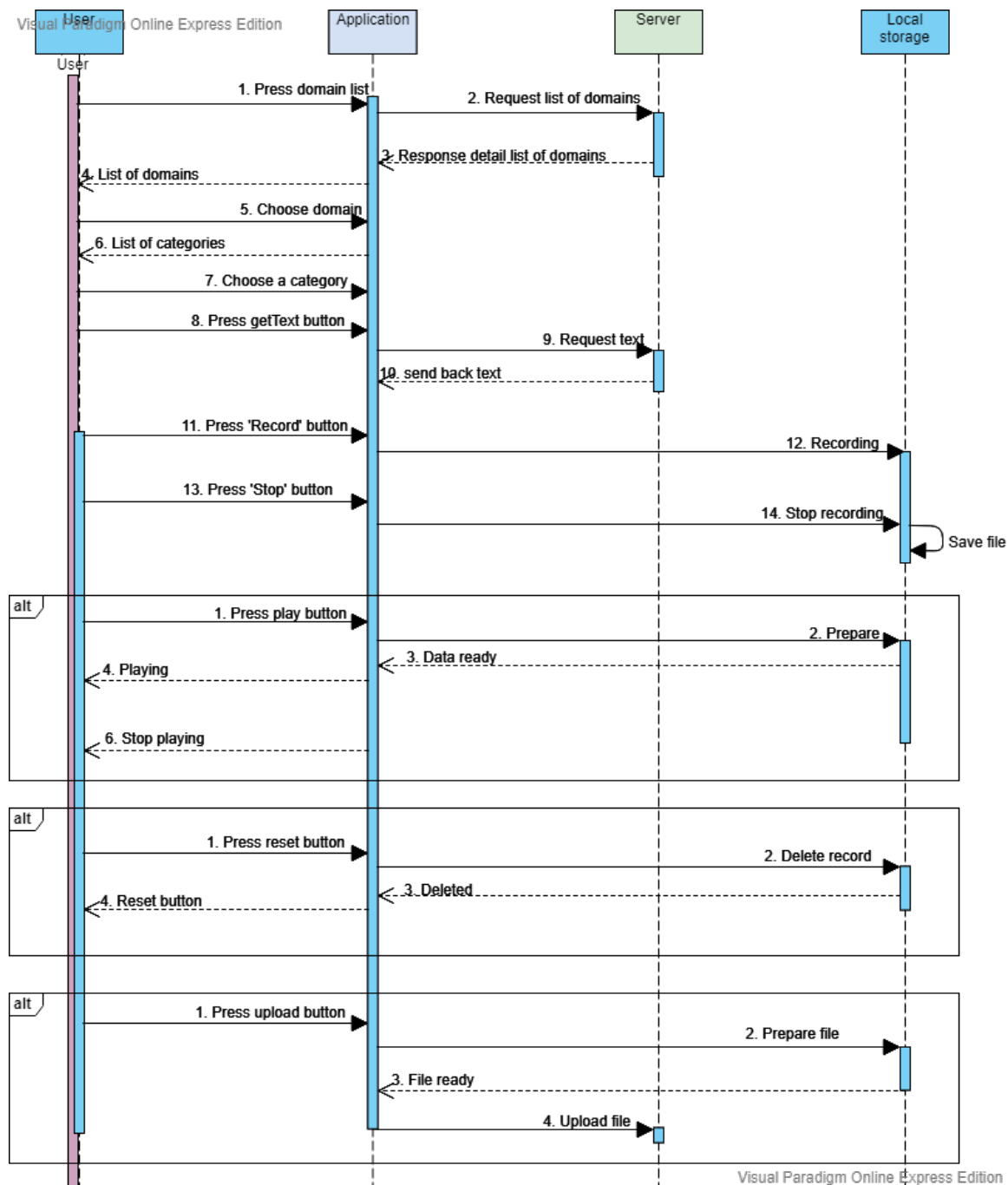


Figure 5: Sequence diagram for Recording and Uploading function

- **Figure 6:** Represent the activities in the Record Evaluation function.

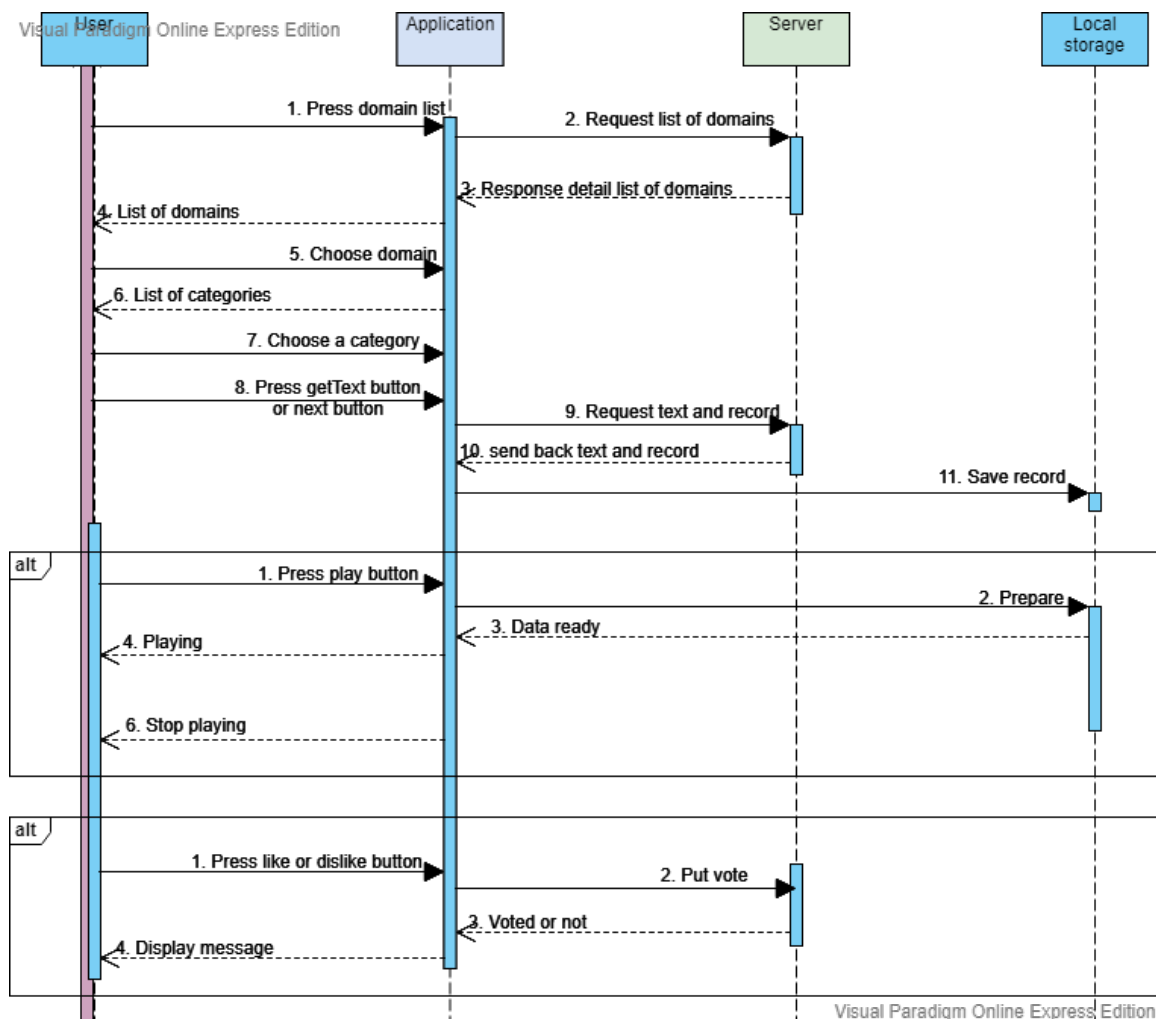


Figure 6: Sequence diagram for Voting function

- **Figure 7:** Explain the action in the Edit paragraph segment function.

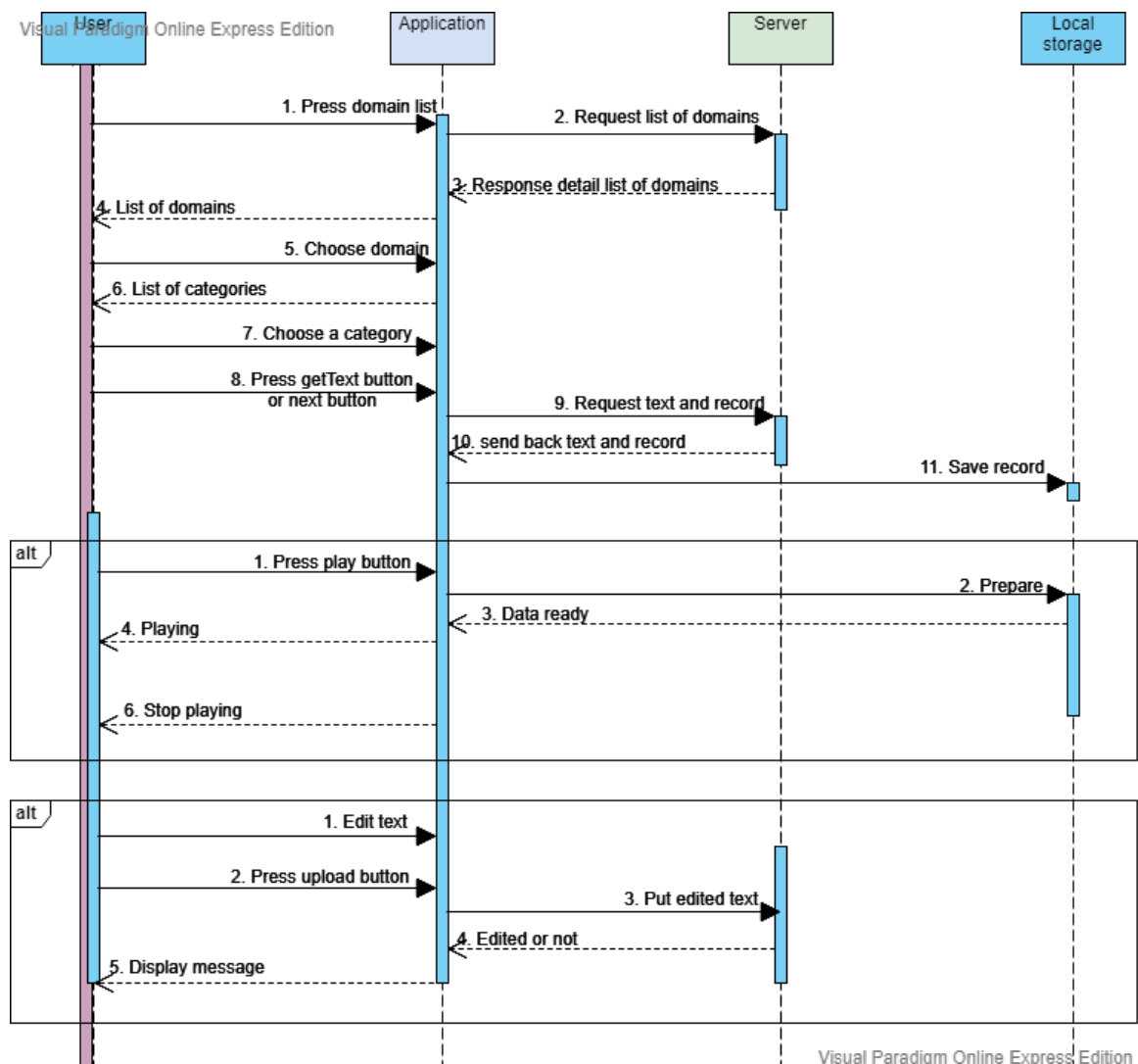


Figure 7: Sequence diagram for Editing function

2.5 Tools and implementation:

2.5.1 Client-Server Model

Client-Server model is a communication model where the server serves requests from the client. With this model, child computers are acting as a client, they are responsible for sending requests to servers then the server handle the request and return the result to that client via network according to figure below. This model allows us to communicate more effectively. Nowadays, client-server model is popularly used everywhere to build network applications since this model allows us to communicate more effectively. In this project, my mobile application can be considered as a network application. Thus, I build my application with the client-server model as a baseline. In the next sections, I will introduce the client-server tools I studied and used in this project.

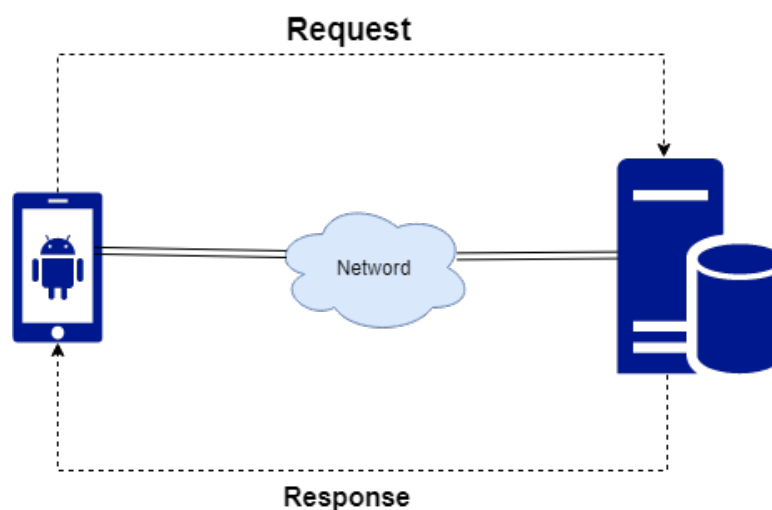


Figure 8: Client-Server Model

2.5.2 Server-side In my system, the server communication with the client via Restful-APIs. To understand more clearly, I will give the definition about some terms. An application programming interface (API) is an interface or communication protocol between a client and a server intended to simplify the building of client-side software. It has been described as a “contract” between the client and the server, such that if the client makes a request in a specific format, it will always get a response in a specific format or initiate a defined action. REST stand for REpresentational State Transfer. REST is web standards based architecture and uses HTTP Protocol. It revolves around resource where every component is a resource and a resource is accessed by a common interface using HTTP standard methods. Software applications written in various programming languages and running on various platforms

can use web services to exchange data over computer networks like the Internet in a manner similar to inter-process communication on a single computer. So, RESTful-APIs are basically API based REST technology. RESTful-APIs are light weight, highly scalable and maintainable and are very commonly used to create APIs for web-based applications. The focus of REST defined using the HTTP method (such as GET, POST, PUT, DELETE ...) and how to format the URL to the application for management of the resource. There are four command used to access the RESTful API:

- **GET:** Provides a read only access to a resource.
- **POST:** Used to create a new resource.
- **DELETE:** Used to remove a resource.
- **PUT:** Used to update an existing resource or create a new resource

2.5.3 Client-side

2.5.3.1 Android platform: About programming language, I choose Java because of its popularity and compatibility. About development environment (IDE), we use Android Studio as the client-side. Android Studio is the official IDE built on JetBrains' IntelliJ IDEA software and specifically designed for Android development. As a consequence, Android Studio supports all the same programming languages of IntelliJ (Java, C++ for example). Therefore, it is one of the most powerful tools for making an android application, which provides plenty of available functions and libraries, so it is easier for us to make our dictionary on android devices.

2.5.3.2 IOS platform: I develop my application for IOS platform with Dart programming language on Flutter SDK. As a cross-platform SDK, Flutter applications can work on both iOS and Android. It is a clever trick to be compatible with the UI framework on both operating systems. Flutter apps do not compile directly with native Android and iOS apps. Instead, they run on the Flutter rendering engine (written in C++) and the Flutter Framework (written in Dart, as well as Flutter applications), both of which are bundled with every application. The reason why did I choose Flutter for the project will be explained in more detail below.

III. Result and Disscusion

3.1 Result

After 6 months, I completed my internship with the following results:

- **On Andorid platform:** There are two application for this platform: one by Java and one by Flutter (Cross-platform, of course!!) The application can work on most android devices with sdk 9.0 and above. Compared to the initial goal, the application has met most of those: Designing the UI, the functions in the original goal are working well. (Apendix A)
- **On IOS platform:** ???

3.2 Discussion

3.2.1 Why flutter?

Why did I choose Flutter over other native programming languages? In fact, I have tried using Objective-C and Swif, however, these two languages are comples to get used to in a short term. Although it is undeniable that the superiority of the native language performance, my application does not need to intervene deeply into the system, so a cross-platform language is completely acceptable. And Flutter has some advangtages that I need:

- **Approachability:** Dart is an object-oriented programming language. It has the concept and syntax similar with Java so it is easy to get acquainted and approach.
- **Reliable:** Dart and Flutter are developed by Google. Although Flutter has only been developed in recent times, with a team of experienced developers, we can completely believe in Flutter's quality and future development.
- **Great performance:** DART is a Statically typed language, so it is AOT (Ahead of Time), then compile before running. Meanwhile, it is also JIT (Just in Time) like dynamic type languages. When developed, it uses JIT to support Hot Load and when build release, use AOT to optimize performance as a normal native code. What about the native module? Unlike JavaScript Bridge, Flutter "talks" to native modules using native interfaces. Although still referred to as a "bridge", it is much faster and almost as "bottlenecked" as React Native.
- **High productivity:** Since Flutter is cross-platform, you can use the same code base for your iOS and Android app. This can definitely save you both time and resources.

3.2.2 Difficult

3.2.2.1 Is the flutter comprehensive? Not yet. Flutter is still in its infancy, and Google has just released the stable version 1.9 on September 10, 2019. In the context of my project, the flutter has revealed certain disadvantages as follows:

- I can't use Native UI Widgets, Flutter is using UI widget build-in, which very wonderful, however some of them still unflexible.
- Packing the entire engine that comes with the application will make the installer bigger. Flutter's FAQ page says that the typical "blank" app is only about 6-7 MB on Android, so no matter how fast the app is, the size increase is a lot.

3.2.2.2 Other difficults As I mention in the sections 2.1, The main data type in the entire system is the voice data that will be stored in a sound file format. The data set will be used to cater for the use of voice as data to train the model. Therefore, there are some specific requirements for the property of the audio file, such as accuracy, the amount of information contained in the file... And in this project, I chose WAV (Waveform Audio File) as an audio file format. WAV file format incorporates lossless compression technique, when it should hold compressed data. Lossless technique means that it can store the audio without compromising in its audio quality even when it holds the compressed data. And this allows the sound of the WAV file to be a close replica of the original audio source. Unfortunately, both Android and iOS platforms do not support direct recording to the WAV format. So I had to write data to the WAV format manually. However, this is still only done with Android versions written in Java.

Responsive is another difficulty. Unlike IOS, the Android ecosystem is much larger, with countless devices and lots of screen sizes. My app may display incorrectly on some devices with screens that are too small or too weird (Like the SamSung Galaxy Fold with a folding screen?!). This is mainly solved by the version written by Flutter.

VI. Conclusion and Future works

All in all, the main components of the application have been successfully implemented:

- **Logging in function:** Users can login successfully to upload their voice record.
- **Recording and Uploading function:** Successfully record the voice of users and send them to the server.
- **Voting function:** Successfully evaluate a record to system
- **Editing function:** Successfully edit and send a text with an audio file if it is not good enough..

To further improve the application in the future, the following tasks need to be done(in order of importance):

1. Improving the performance of the application.
2. Writing audio data in WAV format on IOS.

3. Making application more responsive.
4. Adding new features available in the API (Like translating,...)
5. Improving UI/UX