2017312982 류재현, 2016312445허재승, 2016314568 전형준, 2016310886 김민준, Sesar Hersisson, 위함흔

Introduction to Software Engineering Team 8

REquirement specification

**Table of Contents**

1. Preface 8

a. Objective 8

b. Readership 8

i. User requirements readership 8

i. System requirements readership 8

b. Document structure 8

i. Introduction 8

ii. Glossary 9

iii. User Requirements Definition 9

iv. System Architecture 9

v. System Requirements Specification 9

vi. System Models 9

vii. System Evolution 9

viii. Appendix 10

ix. Index **Error! Bookmark not defined.**

2. Introduction 10

a. Objective 10

b. Needs 10

i. Background: Video Communication Service 10

Figure 1. Online class due to COVID-19 11

ii. Room for improvement, weaknesses 12

c. ATlantis 13

i. Video communication / streaming service 13

ii. Video recording service 13

iii. Video / Audio documentation service 13

iv. Real-time text-based chat service 14

d. Expectations 14

i. Easier search 14

Figure 2. Keyword search with KakaoTalk 14

ii. Reduced burden of stenography 14

3. Glossary 15

a. Objective 15

b. Term definitions 15

Table 1. Term definitions 18

c. Acronyms / Abbreviations 18

Table 2. Acronyms / Abbreviations 18

4. User Requirement Definition 19

a. Objective 19

b. Functional Requirements 19

i. Sign up/Login 19

ii. Video Streaming/Chat 19

Figure 3. Video Streaming / Communication 20

iii. Video Recording 20

Figure 4. Video recording 20

iv. Conversion to script from a video/audio file 20

Figure 5. Google Cloud Speech-to-Text 21

v. Real-time Text-based chat 21

Figure 6. Real-time Text-based chat 21

c. Non-functional Requirements 21

i. Product Requirements 21

1. Performance 21

2. Usability 22

3. Reliability 22

ii. Organizational Requirements 22

1. Implementation requirements 22

2. Equipment requirements 22

iii. External requirements 23

1. Comply with external API terms 23

2. Security 23

5. System Architecture 23

a. Frontend Architecture 23

b. Backend Architecture 23

Diagram 1. Frontend / Backend Architecture 24

Diagram 2. Overall architecture of the system 24

c. Streaming System 25

d. Video Managing System 25

e. Documenting System 25

Diagram 3. Basic flow of the algorithm 26

Diagram 4. Summarization of the overall process 27

6. System Requirement Specification 27

a. External Interface Requirement 27

i. Relations to other I/O 27

ii. Data formats 27

b. Functional Requirement 27

i. Live video and vocal streaming 27

Figure 7. Visualization of video / audio streaming 28

ii. Sound to text conversion 28

iii. Text file linking to voice memo 28

c. Performance Requirement 28

d. Software System Attribute 28

i. Reliability 28

ii. Security 28

iii. Compatibility 29

7. System Model 29

a. Context models 29

i. Context diagram 29

Diagram 5. Overall Context diagram 29

Diagram 6. Overall Process diagram 30

b. Interaction models 30

i. Use-case diagram 30

Diagram 7. Use-case diagram 30

ii. Description table of Use-case diagram 31

1. Login/Register 31

Table 3. Login/Register use-case table 31

2. Video Streaming/Chat 31

Table 4. Video Streaming/Chat use-case table 32

3. View Video 32

Table 5. View Video use-case table 32

4. View Text 32

Table 6. View Text use-case table 33

5. View Real-time Text-based Chat 33

Table 7. View Real-time Text-based Chat use-case table 34

c. Behavioral models 34

i. Documenting System DFD 34

Diagram 8. Data flow diagram for Documenting system 34

8. System Evolution 34

a. Requirements 34

i. System requirements 34

ii. User requirements 35

b. Environment 35

c. Artificial intelligence 35

d. Expected problems 36

i. Problems 36

e. New features 36

i. Screen sharing 36

9. Appendix 36

a. Hardware requirements 36

b. Database requirements 36

i. User 37

ii. RoomInfo 37

iii. Chat 37

# Preface

## Objective

Here, the expected readership of the document is defined, along with a brief explanation of each section of the document. Version history and a summary of changes made in each version are also included.

## Readership

### User requirements readership

Our expected readership is the user of the system in User requirements.

Thus, in order to organize and deliver a clear message of what requirements and definition, we exclude using technical terms, but rather trying to use plain terms.

### System requirements readership

This includes a lot of systemic constraints and requirements which would be read when the process of actual development or it expects to be shown during the contract with the customers.

## Document structure

### Introduction

The motivation of the system to be developed and the background to back up our motivation is introduced. The needs of this system and its expectations are also described.

### Glossary

The technical terms used in this document is organized to make sure this document is clear to any reader, including readers without any expertise in the field, and avoid any misapprehensions that could be made.

### User Requirements Definition

The services provided by the system for the user are described. Both functional and non-functional requirements are described. Simple diagrams may be used along with natural language to provide better understandings for customers.

### System Architecture

An abstract overview of the system’s architecture and the distributions of functions across each sub-system or components are described.

### System Requirements Specification

The user requirements defined in a previous section is specified. Functional and non-functional requirements are mainly explained in more detail.

### System Models

The relationships between the system components and the system and its environment is described, including a graphical visualization.

### System Evolution

The system’s anticipated changes to be made, problems that may occur, or any limits that the system has is explained, so that future system designers can cope with them when it occurs.

### Appendix

Detailed, specific information related to the system to be developed is provided.

# Introduction

## Objective

Here, the background of and the needs of this system is explained and introduced, along with brief descriptions of the system’s functions and how it works, and how it achieves the system’s objectives.

## Needs

### Background: Video Communication Service

Communication systems have evolved throughout history ever since the invention of language and characters. The primitive mankind used face-to-face communication via voice, and thanks to the invention of characters and papers, letters and mails have become the following new form of communication, enabling long-distance communications. This form of communication has been dominant for centuries, until telephones were invented. Telephones have become a new standard for long-distance communications. While letters and mails had the shortcoming that conversation was only available through words, rather than voice, and that it could not be done instantly. Telephones have overcome this, and fulfilled people’s wants to hear other people’s voices, and wants immediate conversation. However, telephones also could not perfectly fulfill people’s demands. Real-time voice communication was satisfying, but was not a suitable method in two aspects: face-to-face communication, and communication with multiple people simultaneously. People demanded services allowing face-to-face communication, and services allowing communication with multiple people, which is eventually for purposes like a meeting, conference, lecture, or a class. This was resolved via video communication services like Zoom, or WebEx. At the same time, another form of communication, text-based real-time chatting service have appeared, and became a prevalent form of communication.

Due to the recent COVID-19, the use of online video communication has skyrocketed, and people’s reliance and importance have reached its peak. It is undoubtedly the state of the art form of communication.

However, as the previous form of communications did, this will not last forever and will have to evolve.

##### 

##### Figure 1. Online class due to COVID-19

### Room for improvement, weaknesses

Current video communication services may seem to be perfect, but also do have some weaknesses. The difficulty of documentation is a critical one. This may seem to be a minor issue, but in fact, it may be quite irritating and a big issue at times. Nowadays, the importance of records is very high, and documentation is the most prevalent form of records. In terms of keeping records, videos can just simply be recorded, but this is not enough in numerous circumstances.

Documentation is important in that it is the basic form of records; documentation is done anywhere possible. For instance, at a court, there is a stenographer who writes down everything said in the court for official documentation. Intuitively, it would be much easier to just record the whole trial as a video. However, stenographer still exists, since documentation is very important, and a video by itself is hard to be considered a valid documentation.

Documentation is important, but also it helps work to be done much more efficiently. After a meeting, one may want to reference the video to check the discussions, like what each person said, what the conclusion was, etc. However, it is not easy to look up, since one has to watch or go through the whole video to find the part where one is looking for. In documents, one can look up for keywords, and look through the parts of a document where they want to see only, but in a video, it cannot be done that easily. Text-based chatting service such as Kakaotalk is much more preferable in this aspect. Text-based chat service and video communication services each have characteristics compensating each other, and a service with both properties, text and video, would be a perfect match for people’s needs.

Therefore, not only recording the video itself, but also having the speech recorded automatically, this critical issue of documenting video communication systems can be overcome.

## ATlantis

Our service, ATlantis is a video communication system, mainly serves the following features.

### Video communication / streaming service

ATlantis is mainly a video communication system, where users can stream their live videos and communicate with others, for purposes like a conference, meeting, lecture, class, etc.

### Video recording service

This is a very basic functionality, which is necessary for a video communication / streaming service. When a video is streamed live, it can be recorded at the same time if the user wishes to do so. The main screen of the video recorded will be the screen of the user’s perspective.

### Video / Audio documentation service

* + - 1. ATlantis’ distinctive feature: the video can be documented real-time during a communication is on-going, into a text-based chat. The contents, “who” said “what” and “when” will be recorded.
      2. Not only can this feature be used live, but it can also be used afterwards, with a recorded video or an audio file. Users may input a video / audio file recorded from another platform.

### Real-time text-based chat service

This is also another very basic functionality, also existent in current video communication systems. Not only is a communication done through video, but also through

## Expectations

### Easier search

All the dialog will be saved in texts, which will make keyword-based search available. With this privilege, users can watch or read the specific part of the conversation again, even after the whole meeting / class / conversation is finished.

##### 

##### Figure 2. Keyword search with KakaoTalk

### Reduced burden of stenography

If a meeting / class / conversation has to be written in text, usually a person is allocated to do the job. However, this service will reduce the burden to do so, and the automatic system will do the job on behalf of a person. This would mean more people can be put into other works and reduce the overall cost in work.

# Glossary

## Objective

Here, the terms used throughout this document are explained and defined. In case there is a reader without any background knowledge of the field, terms are to be defined as easily as possible, and clearly so that there are no misunderstandings to be made.

## Term definitions

|  |  |
| --- | --- |
| Term | Definition |
| Algorithm | A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer. |
| Application Programming Interface | A computing interface which defines interactions between multiple software intermediaries. It defines the kinds of calls or requests that can be made, how to make them, the data formats that should be used, the conventions to follow, etc. |
| ASF | Microsoft’s digital audio/digital video container format, especially meant for streaming media. |
| Backend | The data access and processing layer of a piece of software, the parts of a software that cannot be explicitly seen. |
| Database | An organized collection of data, generally stored and accessed electronically from a computer system. |
| Frontend | The presentation layer of a piece of software, the parts of a software that can be seen explicitly. |
| F1 score | A measure to calculate a test’s accuracy. |
| Git | A distributed version-control system for tracking changes in source code during software development. |
| Cloud | The on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. |
| Graphics Processing Unit | A specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. |
| Identifier | A user identifier to a human being or to another computer or network component. In computer systems, specific identifiers need to be linked to particular authorized users of those identifiers. |
| Interface | A device or a system that unrelated entities use to interact. |
| MP3 | A coding format for digital audio |
| Operating System | The software that supports a computer's basic functions, such as scheduling tasks, executing applications, and controlling peripherals.  ex) Windows, Mac, Linux, etc. |
| Password | A memorized secret, typically a string of characters, used to confirm the identity of a user. |
| Server | A computer program or a device that provides functionality for other programs or devices |
| Stream | The process of constantly receiving and obtaining media while being delivered by a provider. |
| User Interface | The means by which the user and a computer system interact, in particular the use of input devices and software. |
| WAV | An audio file format standard, developed by Microsoft and IBM, for storing an audio bitstream on PCs. |

##### Table 1. Term definitions

## Acronyms / Abbreviations

|  |  |
| --- | --- |
| API | Application Programming Interface |
| DB | Database |
| GPU | Graphics Processing Unit |
| ID | Identifier |
| I/O | Input / Output |
| OS | Operating System |
| PW | Password |
| UI | User Interface |

##### Table 2. Acronyms / Abbreviations

# User Requirement Definition

## Objective

Here, the services ATlantis provides for the users are described. Both functional requirements and non-functional requirements are to be described, in natural language along with diagrams for further explanation. The system is to fulfill the mentioned requirements.

## Functional Requirements

### Sign up/Login

Sign up/Login is required to use features such as video chat and Real-time Text-based chat. Users must enter their ID, PW, name and date of birth for Sign up/Login. The ID must be unique, and other information may be duplicated. After joining the service, users can log in using the ID and PW entered at the time of signing up.

### Video Streaming/Chat

It is divided into a video streaming function that provides video content by one user and a video chat function that allows multiple users to chat on voice/video. Both features allow one user to create a chat room and share a link in the chat room so that other users can join the chat. In the case of video streaming, users who create rooms can deliver content via video, while those who participate can deliver their opinions through text chat. For video chat, the user who created the room and the user who participated have the same rights, and each user can chat by voice. At this point, the screen is printed as a video of one of the participating users.



##### Figure 3. Video Streaming / Communication

### Video Recording

This function automatically saves video when Video Streaming /Chat is performed. When Video Streaming/Chat is in progress, it is divided into video files and audio streams. A video file can be used later when the user wants to check Video Streaming/Chat again, and an audio stream is used to convert audio to text.



##### Figure 4. Video recording

### Conversion to script from a video/audio file

A function that automatically converts the contents of a meeting into text based on the audio stream stored through video recording, which is the most important function provided by the ATlantis These features will be implemented using Google Cloud Speech-to-Text.

****

##### Figure 5. Google Cloud Speech-to-Text

### Real-time Text-based chat

It is a function that shows the results of Conversion to script from a video/audio file in the form of a chat. Voice recognition will allow users to recognize who said each text and show users the same results that the person would enter text chat.

##### 

##### Figure 6. Real-time Text-based chat

## Non-functional Requirements

### Product Requirements

#### Performance

As it is a real-time chat, the gap between the input and output should not be long. In comparison, converting Audio to text does not have to be done in real time, so the available resources will be allocated first to real-time chat and then to text conversion after the real-time chat is over.

#### Usability

Due to the nature of the system with chat function, if the method of use is difficult, the merit of use is reduced. Therefore, a clean UI and clear documentation will be provided for easy use so that users are not confused about their use.

#### Reliability

Care should be taken not to compromise the accuracy of the conversion of Audio. The accuracy of text to audio function will be measured using F1 score. Since there is not much time given, there is no time to retrain the model even if the F1 score is not good, but it will be meaningful to measure the model's performance.

### Organizational Requirements

#### Implementation requirements

Git will be used for smooth version management as several people develop at the same time. In addition, development plans will have to be clearly formulated for parallel development. It will also need to learn the manual to use the Google Cloud Speech-to-text API.

#### Equipment requirements

A GPU is likely to be needed to use Google Cloud Speech-to-Text. If conditions allow, it will use a server equipped with GPU, otherwise the audio to text function will be slow.

### External requirements

#### Comply with external API terms

Because we use external APIs, we should keep the terms and conditions of those APIs accurately. For example, the API used is prohibited from commercial use, and if we use it to make a profit, you may have to take legal responsibility.

#### Security

The contents of the chat should not be viewed by unauthorized users. To prevent this, the deadline for an open chat room invitation link will be limited to one week. In addition, the service will be unavailable if login is not performed to prevent reckless contact from outside.

# System Architecture

## Frontend Architecture

* + 1. Frontend architecture is equal to the user interface. In Atlantis, the main interface is made of Start Video Streaming/Chat, Saved Video, Audio Documenting, See Real-time text-based Chat.

## Backend Architecture

* + 1. Backend System consists of Application Server, Streaming System, Video Managing System, Documenting System.

##### 

##### Diagram 1. Frontend / Backend Architecture

##### 

##### Diagram 2. Overall architecture of the system

## Streaming System

* + 1. Streaming System provides streaming services Making new room, Inviting Users, Streaming real-time Video / Voice of multiple users. And it gives videos’ information to Video managing System

## Video Managing System

* + 1. Video Managing System saves whole streams and stores video of streams in Video DB. It extracts audio from video and stores audio in Audio DB.

## Documenting System

* + 1. Documenting System gets audio from Audio DB to Google Cloud for using Speech-to-text API and server delivers user’s information and documented text to Text DB. Another function shows Real-time Text-based Chat through Using Speech-Recognition-Algorithm with text in Text DB.

##### 

##### Diagram 3. Basic flow of the algorithm

##### 

##### Diagram 4. Summarization of the overall process

# System Requirement Specification

## External Interface Requirement

### Relations to other I/O

Input and output is occured in each of the user's local device, natural language recording of input device’s microphone and it’s graphic , speaker will be I/O devices.

### Data formats

Since we use voice recording and support live streaming, ASF is needed for video streaming, and WAV or MP3 for voice streaming and recording.

## Functional Requirement

### Live video and vocal streaming

User’s voice and video(via locally attached camera) should be streamed through all users participating in a certain group. (e.g. broadcast) If there is an unexpected delay due to the sudden high traffic, system should drop down the quality of video and sound so that participants have to be guaranteed the real time services.

##### Figure 7. Visualization of video / audio streaming

### Sound to text conversion

System should convert dialog voice memo into text file. Natural language processing is required in this functionality.

### Text file linking to voice memo

System should determine certain keywords which make a link to voice memo time location. (e.g. If a user clicks that highlighted keyword, voice memo can be played at which that keyword is spoken.)

## Performance Requirement

Long delay should be avoided.

## Software System Attribute

### Reliability

All users have to get all the streaming videos and sounds without any packet drops. And the recording storage has to be replicated in order to level up storage reliability.

### Security

Videos and sounds should not be leaked.

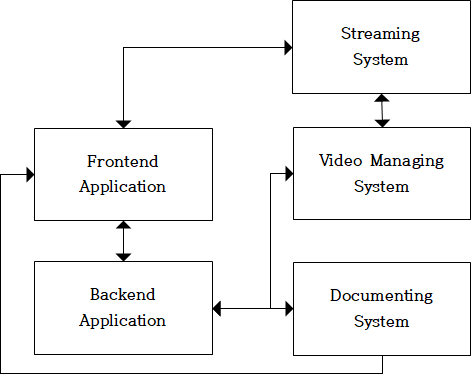
### Compatibility

It should be compatible with multiple OSes.

# System Model

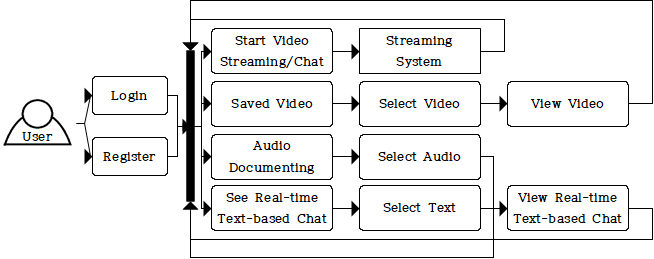
## Context models

### Context diagram

****

##### Diagram 5. Overall Context diagram

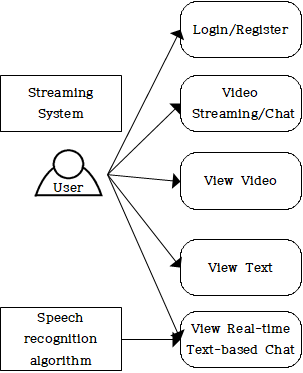
* + 1. **Process diagram**

****

##### Diagram 6. Overall Process diagram

## Interaction models

### Use-case diagram

****

##### Diagram 7. Use-case diagram

### Description table of Use-case diagram

#### Login/Register

|  |  |
| --- | --- |
| **Actor** | User, Database(User) |
| **Description** | The process of identifying connected user with the information in DB is same or not |
| **Action** | User inputs ID and PW |
| **Success Response** | User’s ID exist in DB and Input PW is same in User DB’s data, give access rights |
| **Failure Response** | If User’s ID doesn’t exist in the DB, move to the Register Section. If ID exist but PW doesn’t correct, pop up error message : PW Incorrect |

##### Table 3. Login/Register use-case table

#### Video Streaming/Chat

|  |  |
| --- | --- |
| **Actor** | Users, Streaming System |
| **Description** | Live Streaming/Chat Service of Multiple Users |
| **Action** | User select Start Streaming in main page |
| **Success Response** | If connections between users are clear, it can start streaming or make another room for others |
| **Failure Response** | If connections aren’t clear, pop up error message: ‘Connection error’ |

##### Table 4. Video Streaming/Chat use-case table

#### View Video

|  |  |
| --- | --- |
| **Actor** | Users, DB(Video) |
| **Description** | User can view saved video |
| **Action** | User select ‘Saved Video’ in main page |
| **Success Response** | Show list of saved Videos while streaming |

##### Table 5. View Video use-case table

#### View Text

|  |  |
| --- | --- |
| **Actor** | Users, DB(Audio, Text), Documenting System |
| **Description** | User can view text which is documented video |
| **Action** | In ‘Audio documenting’ select audio |
| **Success Response** | Show the text of selected audio and save it in DB |
| **Failure Response** | If selected audio isn’t available type or damaged, pop up error message : Not available type or damaged |

##### Table 6. View Text use-case table

#### View Real-time Text-based Chat

|  |  |
| --- | --- |
| **Actor** | User, DB(Text), Documenting System(Speech recognition Algorithm) |
| **Description** | Change text to Real-time Text-based Chat by using Speech recognition Algorithm |
| **Action** | Real-time Text-based Chat in main page, select text file |
| **Success Response** | If file is fine and algorithm works, print the text file into Real-time Text-based Chat |
| **Failure Response** | If file is not available, pop up error message : Not available file |

##### Table 7. View Real-time Text-based Chat use-case table

### Behavioral models

### Documenting System DFD

* + - 1. Make the behavioral model of the main part which is Documenting System

##### Diagram 8. Data flow diagram for Documenting system

# System Evolution

## Requirements

### System requirements

As the user-base of the software grows the system has to be maintained and scaled accordingly. The server that serves the requests has to be upgraded in a way it can handle the increasing amount in requests and serve all users.

### User requirements

The system should not be scaled in such a way that it requires significantly more compute power on the user-end. The application should be able to run on older computers but then again should not be maintained for computers that have been outdated long ago.

## Environment

* + 1. As technology gets better, especially network speeds, the video and audio stream should be updated accordingly. Higher network speeds mean that higher quality video and audio can be streamed without interruption.
    2. Server engineering and runtime environments also keep getting better allowing larger applications to be run on lower compute power. The system should keep up to date with the latest in these areas to maximize efficiency and quality of the application.

## Artificial intelligence

* + 1. The system utilizes artificial intelligence in its speech to text recognition which will continue to improve as time goes on and it processes more data.

## Expected problems

### Problems

Video and audio streaming can be hard to optimize for different OS and hardware and we could run into problems after the system’s release where for example a certain hardware or a version of OS will not be able to run the application, which therefore have to be fixed.

## New features

### Screen sharing

Most video and audio streaming services support screen sharing. This is something that should be implemented in our application after its release.

# Appendix

## Hardware requirements

We will use web server development frameworks such as Django and node because we will provide services based on web servers. And we need a server that can connect to the network because we have to provide a video chat service. When converting Voice to text, we will use the Google cloud text to speech API, which requires a lot of operations, so the server must be equipped with a GPU. Specifically, it would be better to use GPUs with performance of GeForce GTX 1060 or higher.

## Database requirements

ATlantis does not have a complex database structure because the database is not the main service. The database will be managed using a DBMS that fits the web framework to be used. The brief structure of the database is as follows. The tuple corresponding to PK is underlined.

### User

It is an entity that contains user information.

Tuple: ID, PW, Name, Birthday

### RoomInfo

It is an entity that contains information about chatting room, streaming room.

Tuple: RoomNum, RoomCategory, URL, RoomUserID, ChatInfo, Admin

### Chat

An entity that contains information about each chat.

Tuple: ChatNum, ChatContent, ChatUserID