



# **VRoom**

## **Virtual Meeting Room Service**

Software Requirement Specification

2021.10.31.

### **Introduction to Software Engineering 41**

#### **Team 2**

Team Leader	Kim Jinwoong
Team Member	Hong Seongjun
Team Member	Kim Hongbeen
Team Member	Kim Minhee
Team Member	Lee Hyewon
Team Member	Lee Jiwoo
Team Member	Seo Chaeyeon

## CONTENTS

<b>1. INTRODUCTION .....</b>	<b>7</b>
1.1. Purpose .....	7
1.2. Scope .....	7
1.3. Definitions, Acronyms, and Abbreviations .....	7
1.4. References .....	9
<b>2. OVERALL DESCRIPTION .....</b>	<b>10</b>
2.1. Product Perspective .....	10
2.1.1. System Interfaces .....	10
2.1.1.1. Pen.....	10
2.1.1.2. Projector.....	10
2.1.1.3. Calendar .....	10
2.1.1.4. Voice chat .....	10
2.1.1.5. Whisper Zone.....	10
2.1.1.6. Avatar .....	10
2.1.2. User Interfaces.....	10
2.1.2.1. Common Interfaces.....	11
2.1.2.2. Host Interfaces.....	11
2.1.2.3. Participants Interfaces .....	11
2.1.3. Hardware Interfaces.....	11
2.1.4. Software Interfaces .....	11
2.1.4.1. Unity (v.2019.4.30f1).....	11
2.1.4.2. VRChat .....	11
2.1.5. Memory Constraints .....	11
2.2. Product Functions.....	11
2.2.1. Pen.....	11
2.2.2. Projector.....	12
2.2.3. Calendar.....	12
2.2.4. Voice chat.....	12
2.2.5. Whisper Zone .....	12
2.2.6. Avatar .....	12
2.3. User Characteristics .....	12
2.3.1. Manager .....	12
2.3.2. Host.....	13

2.3.3. Participant .....	13
2.4. Constraints .....	13
2.5. Assumptions and Dependencies .....	14
<b>3. SPECIFIC REQUIREMENTS .....</b>	<b>15</b>
3.1. External Interface Requirements .....	15
3.1.1. User Interfaces .....	15
3.1.2. Hardware Interfaces .....	17
3.1.3. Software Interfaces .....	18
3.1.4. Communication Interfaces .....	19
3.2. Functional Requirement .....	19
3.2.1. Use Case .....	19
3.2.2. Use Case Diagram .....	22
3.3. Performance Requirements .....	22
3.3.1. Static Numerical Requirements .....	23
3.3.2. Dynamic Numerical Requirements .....	23
3.4. Design Constraints .....	23
3.5. Standards Compliance .....	23
3.5.1. Compliance for Submitting World .....	23
3.5.2. Compliance for Performance .....	24
3.5.3. Compliance for General .....	24
3.6. Software System Characteristics .....	24
3.6.1. Product Requirements .....	24
3.6.1.1. Usability Requirements .....	24
3.6.1.2. Performance Requirements .....	25
3.6.1.3. Reliability Requirements .....	25
3.6.1.4. Security Requirements .....	25
3.6.2. Organizational Requirements .....	25
3.6.2.1. Operational Requirements .....	25
3.6.3. External Requirements .....	25
3.6.3.1. Safety / Security Requirements .....	25
3.6.3.2. Regulatory Requirements .....	26
3.7. Organizing the Specific Requirements .....	26
3.7.1. Class Model .....	26
3.7.2. Object Model .....	26
3.7.3. Interaction Model .....	27
3.7.4. Behavior Model .....	27

---

3.7.4.1. Sequence Diagram .....	27
3.8. System Architecture.....	28
3.9. System Evolution.....	28
3.9.1. Limitation and Assumption .....	28
3.9.2. Evolutions of Hardware and Change of User Requirements .....	28
<b>4. SUPPORTING INFORMATION .....</b>	<b>30</b>
4.1. Document History .....	30

## LIST OF FIGURES

[Figure 1] Use Case Diagram .....	22
[Figure 2] Class Model .....	26
[Figure 3] Object Model .....	27
[Figure 4] Sequence Diagram .....	27
[Figure 5] System Architecture .....	28

## LIST OF TABLES

[Table 1] Table of Terms and Definitions .....	8
[Table 2] Table of Acronyms and Abbreviations .....	8
[Table 3] User Interface of Basic User Interaction .....	15
[Table 4] User Interface of Host Panel .....	15
[Table 5] User Interface of Participant Panel .....	16
[Table 6] User Interface of Interaction Panel.....	16
[Table 7] User Interface of Projector Panel.....	17
[Table 8] User Interface of Calendar Panel.....	17
[Table 9] Hardware Interface of VRoom.....	18
[Table 10] Software Interface of VRoom Use: Unity .....	18
[Table 11] Software Interface of VRoom Use: VRChat .....	18
[Table 12] Software Interface of VRoom Development: Unity.....	18
[Table 13] Software Interface of VRoom Development: VRChat SDK.....	19
[Table 14] Software Interface of VRoom Development: VRChat Udon.....	19
[Table 15] Use Case of Instantiate.....	19
[Table 16] Use Case of Join.....	20
[Table 17] Use Case of Pen & Eraser .....	20
[Table 18] Use Case of Projector .....	20
[Table 19] Use Case of Calendar .....	21
[Table 20] Use Case of Whisper zone .....	21
[Table 21] Use Case of Seat .....	21
[Table 22] Use Case of Avatar .....	21
[Table 23] Document History.....	30

# 1. Introduction

## 1.1. Purpose

This document is a Software Requirements Specification (SRS) for the online virtual meeting room service, VRoom. The purpose of this document is to outline the requirements for the service. All features of the service are designed and implemented by Team 2 of the Introduction to Software Engineering course (2021 Fall) at Sungkyunkwan University in accordance with the content described in this document.

The main readers of this document are all members of Team 2, professor and TAs of this course, and other students attending this course. This document serves as a basis for design and implementation for Team 2 and serves as a manual for other readers to find the purpose and features of the service.

## 1.2. Scope

VRoom is a virtual meeting room service provided on the VRChat platform. Users can select their avatars and gather in a virtual meeting room to communicate. Voice chat is provided as a basic means of communication. As an add-on to voice chat, there is a special zone called the whisper zone, where conversations between people inside are not heard from outside. For more effective communication, a projector for presenters to share their screens and a pen for writing on any wall are provided. In addition, the calendar feature makes it possible to share schedules during meetings. Only permitted persons can enter the conference room, and all communication in the conference room is kept private.

VRoom can be used for any kind of meeting, such as a group assignment meeting, group study, business presentation, etc. Because it does not stream the appearance of meeting participants, it is free from the Zoom fatigue problem, which refers to the stress of participants during a video call from being constantly conscious of how they will appear on the screens of other participants. In addition, since it is implemented in a 3D environment, it provides superior immersion and realism than the existing video call services. It is expected to further develop the telecommuting culture triggered by the pandemic.

## 1.3. Definitions, Acronyms, and Abbreviations

The following table defines terms used in this document.

[Table 1] Table of Terms and Definitions

Terms	Definitions
Avatar	An animation character that a user presents as a substitute for the user's role in VRoom world
First-person view	A view on the user's screen that shows only surrounding environment
Host	The person who hosts the meeting
Maintenance	Fixing software products that correct defects and improving performance and other characteristics after deployment
Manager	A person who manages the VRoom system
Participants	Meeting participants except for the host
Reliability	Probability of failure-free operation of a computer program for a specified period in a specified environment
Scalability	Property of a system to handle a growing amount of work by adding resources to the system
Software	One or more computer programs for a specific purpose on storage devices
User	People who use the system
VRChat	Virtual reality-oriented voice chat software developed by VRChat Inc. in the US
VRoom world	An instance of VRoom which means a single meeting room

The following table explains the acronyms and abbreviations used in this document.

[Table 2] Table of Acronyms and Abbreviations

Acronyms & Abbreviations	Explanation
Metaverse	It is a new word that combines 'meta' and 'universe' and refers to a virtual world where real life and legally recognized activities such as job, finance, and learning are connected in 3D
SDK	Software development kit, which refers to a set of software-building tools for a specific platform



## 1.4. References

- "IEEE Recommended Practice for Software Requirements Specifications," in IEEE Std 830-1998, vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286.
- Team 1 of the Introduction to Software Engineering course at Sungkyunkwan University (2020) iDecide [SRS Document].  
[https://github.com/skkuse/2020spring\\_41class\\_team1/blob/master/docs/SRS\\_TEAM1.pdf](https://github.com/skkuse/2020spring_41class_team1/blob/master/docs/SRS_TEAM1.pdf)
- "SDK vs. API: What's the difference?" IBM. [Online]. Available:  
<https://www.ibm.com/cloud/blog/sdk-vs-api>. [Accessed: 28-Oct-2021].
- V. Mallawaarachchi, "10 common software architectural patterns in a Nutshell," Medium, 02-Sep-2020. [Online]. Available:  
<https://towardsdatascience.com/10-common-software-architectural-patterns-in-a-nutshell-a0b47a1e9013>. [Accessed: 28-Oct-2021].
- VRChat Documentation.  
<https://docs.vrchat.com>

## **2. Overall Description**

### **2.1. Product Perspective**

#### **2.1.1. System Interfaces**

##### **2.1.1.1. Pen**

The pen is located in front of projector. All users can draw using the pen object. By using this, it is possible to visually check the contents of the entire meeting, and it helps in communication and visualization between participants.

##### **2.1.1.2. Projector**

Projector is located on one side of wall. By sharing files such as PDF or PPT in real time between meeting participants, it is possible to facilitate the progress of the meeting.

##### **2.1.1.3. Calendar**

A calendar is placed in the VRoom to help schedule the project. It is always accessible through interaction, and you can use this feature to help manage project progress and check the schedule.

##### **2.1.1.4. Voice chat**

Because VRoom is a meeting space, it provides advanced voice chat. As a basic function provided by VRChat, both host and participants can use the function through the microphone and communicate freely.

##### **2.1.1.5. Whisper Zone**

VRoom supports whisper zone where mute function exists for voice chat. Users can use it directly on itself by entering it.

##### **2.1.1.6. Avatar**

VRoom provides suitable clothes and accessories for the meeting situation so that users can decorate their avatar.

#### **2.1.2. User Interfaces**

### **2.1.2.1. Common Interfaces**

- Use the pen object in front of the projector
- Share display with projector features
- View participants
- Mute on oneself
- Sit on seats in the meeting room

### **2.1.2.2. Host Interfaces**

- Create a password for the meeting room and start the meeting
- Mute on other participants

### **2.1.2.3. Participants Interfaces**

- Join the meeting by entering the password generated by the host

## **2.1.3. Hardware Interfaces**

The system is intended to run in VRChat as a Steam application in PC. The PC must have 4GB RAM and Windows 8.1 or Windows 10.

## **2.1.4. Software Interfaces**

### **2.1.4.1. Unity (v.2019.4.30f1)**

It is essential software to develop all functions of the system.

### **2.1.4.2. VRChat**

It is the platform that users will actually use and encounter the system.

## **2.1.5. Memory Constraints**

The system should run on PC with least 4GB RAM memory and the system requires at least 1GB for installation and execution.

## **2.2. Product Functions**

### **2.2.1. Pen**

- Draw 2D freely using the pen

- Show the drawings on the wall where all participants can see what is drawn

### **2.2.2. Projector**

- Start/end screen sharing
- Change the projected screen back and forth

### **2.2.3. Calendar**

- Show today's calendar
- Flip the calendar back and forth by month
- Create schedule

### **2.2.4. Voice chat**

- Communicate with other participants using voice

### **2.2.5. Whisper Zone**

- A zone where the sound is bound to it, muting one's voice inside it

### **2.2.6. Avatar**

- Change avatars in the provided closet

## **2.3. User Characteristics**

### **2.3.1. Manager**

Manager in this document is the person who manages VRoom service. Because our service aims to operate in the VRChat application, the developed software should be managed to operate well in VRChat. Therefore managers do upload and update our service to VRChat as one “world”. They need to fill in an application form to post the “world” public on VRChat.

Managers are assumed that they should have sufficient knowledge of the VRChat system. Also they should have a good understanding of the concept and examples of “world” in VRChat system. In addition, they should manage requests or recommendations from VRChat.

In order to upload to VRChat, developed software must meet the optimization criteria and performance guidelines suggested by VRChat. Thus managers should be familiar with the information, and review in the process of uploading and updating. And because they need to understand the part of performance, they should major in discipline related with computer or software.

### **2.3.2. Host**

A host in this document is the person who is the host of the meeting in VRChat application. Hosts intend to host meetings in an online metaverse environment. Hosts are assumed that they have VRChat or Steam IDs, and there is an environment in which Steam's VRChat is installed for them. Also, they are assumed that they have microphone and camera environment if they want in their meeting. In addition, English should be available to the extent that there should be no difficulty in installing and understanding the VRChat service. It is because the VRChat service is basically operated in English. Therefore, it is generally assumed that hosts are aged 15 to 60.

### **2.3.3. Participant**

A participant in this document is the person who wants to attend the meeting in VRChat application. Participants intend to enter the meeting room. It is assumed that the meeting room that the participants intend to enter has already been hosted by the host. In addition, participants are assumed that they have the password to access the meeting room was delivered in advance by the host.

Similar to the hosts, they are assumed that they have an environment where VRChat is available and have a basic English level. Therefore, it is expected that participants are aged 15 to 60.

## **2.4. Constraints**

This system will be designed and implemented based on the contents mentioned in this document. Other details will be determined by developers in development phase. The determined contents are as follows:

- Develop according to the VRChat regulations
- Develop to satisfy the optimization criteria and performance guidelines suggested by VRChat

- Do not let the file size of developed software exceed 200MB, because VRChat limits the size of the world to 200MB
- Develop with “Unity 2019.4.30f1” version
- Develop in Windows environment
- Check and review the results or outputs through VRChat in the Steam application
- Use open source if possible
- Consider the scalability of the system (so that we can add more functions in the future)
- Consider system cost and maintenance cost
- Develop to make the system resource be minimized when writing the source code
- Develop the source code in consideration of future maintenance, and write comments if possible
- Decide in the direction of improving system performance and reliability
- Design the system easy for users to use and understand it
- Use software or technology that is already widely used and has a lot of related documents
- Do not use software or technology that may be paid unless it is irreplaceable and necessary
- Do not use software or technology that may expose users' personal information
- Develop the system not to violate users' privacy according to national privacy standard

## **2.5. Assumptions and Dependencies**

This system is designed and implemented in the direction specified in this document. Therefore, all implementations should be developed in Unity 2019.4.30f1 version in Windows environment. And they should be operated in the VRChat in the Steam application of the PC environment. Thus, it may not be applied to other operating systems and environments and versions. If the software or technologies which we want to use do not work on Unity 2019.4.30f1 version, we do not use them and maintain the current version (it means Unity 2019.4.30f1 version).

## 3. Specific Requirements

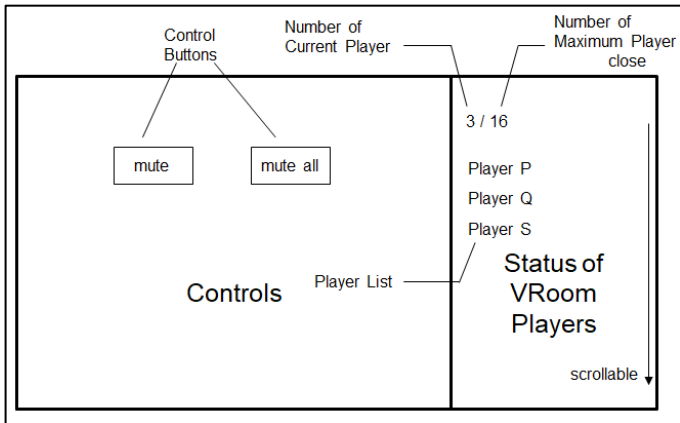
### 3.1. External Interface Requirements

#### 3.1.1. User Interfaces

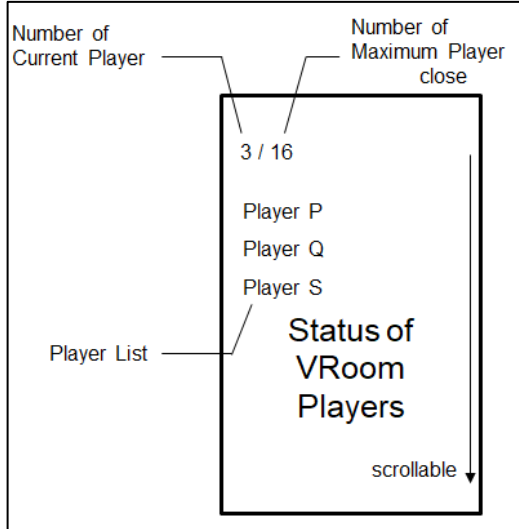
[Table 3] User Interface of Basic User Interaction

Basic User Interaction	
Purpose	To control the user's avatar and interact with objects.
Input/Output	Keyboard and mouse / User Screen
Relationship with other input/outputs	Prioritized before other interfaces.
UI formats and organization	Provided by VRChat.

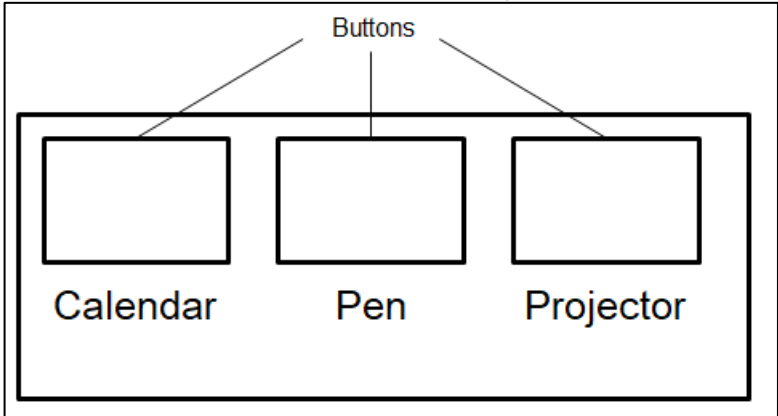
[Table 4] User Interface of Host Panel

Host Panel	
Purpose	To control users' voice and check the status of the VRoom world.
Input/Output	Mouse / Host Panel
Relationship with other input/outputs	Triggered by pushing 'P' key. Closed by clicking 'close' or by pushing 'P' key again.
UI formats and organization	<p>It is located in front of the host at the time triggered. It is fixed at where it is located. It is not visible to other users.</p> 

[Table 5] User Interface of Participant Panel

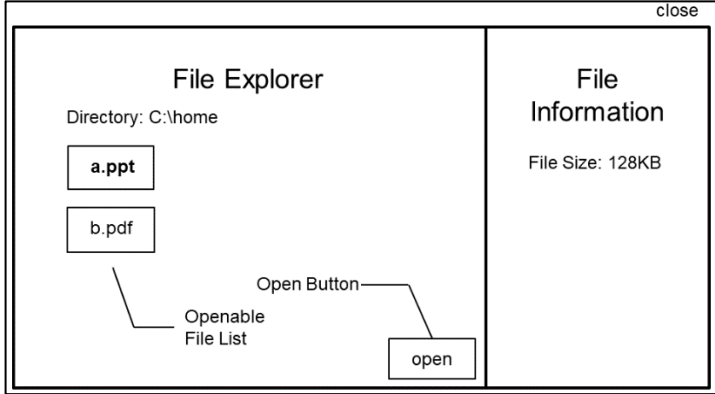
Participant Panel	
Purpose	To check the status of the VRoom world.
Input/Output	Mouse / Participant Panel
Relationship with other input/outputs	Triggered by pushing 'P' key. Closed by clicking 'close' or by pushing 'P' key again.
UI formats and organization	<p>It is located in front of the member at the time triggered. It is fixed at where it is located. It is not visible to other users.</p> 

[Table 6] User Interface of Interaction Panel

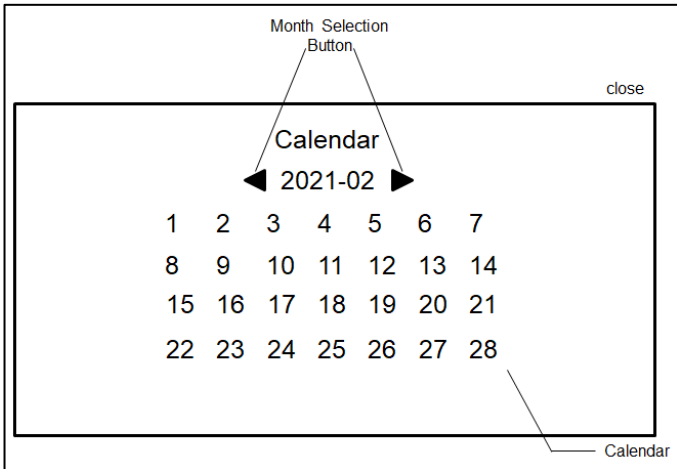
Interaction Panel	
Purpose	To trigger interactions for the world objects.
Input/Output	Mouse / Nothing
Relationship with other input/outputs	It triggers panels(calendar panel, pen panel, projector panel) to open.
UI formats and organization	<p>It is located next to the screen and on the table. It and the access for it can be controlled by the host.</p> 



[Table 7] User Interface of Projector Panel

Projector Panel	
Purpose	To set the projector feature in the screen.
Input/Output	Mouse / Projector Panel
Relationship with other input/outputs	Triggered by clicking 'Projector Button' from 'Interaction Panel'.
UI formats and organization	<p>It is located next to the screen.</p> 

[Table 8] User Interface of Calendar Panel

Calendar Panel	
Purpose	To check dates.
Input/Output	Mouse / Calendar
Relationship with other input/outputs	Triggered by clicking 'Calendar Button' from 'Interaction Panel'.
UI formats and organization	

### 3.1.2. Hardware Interfaces

VRoom must satisfy hardware system requirements of VRChat requirements.

[Table 9] Hardware Interface of VRoom

Unity	
OS	Windows 8.1, 10
Processor	Intel i5-4590 / AMD FX 8350 equivalent or greater
Memory	4GB or greater
Graphic	NVIDIA GeForce GTX 970 / AMD Radeon R9 290 equivalent or greater
Internet	Required
Storage	1GB or greater

### 3.1.3. Software Interfaces

For VRoom use, Unity and VRChat are used as a software interface.

[Table 10] Software Interface of VRoom Use: Unity

Unity	
Purpose	To support the VRChat platform.
Relationship with other input/output	It is required for VRChat.
Version	Unity 2019.4.30f1

[Table 11] Software Interface of VRoom Use: VRChat

VRChat	
Purpose	To support VR and 3D world functionalities.
Relationship with other input/output	It is required for VRoom.
Version	2021.3.4

For VRoom development, Unity, VRChat SDK and VRChat Udon are used as software interfaces.

[Table 12] Software Interface of VRoom Development: Unity

Unity	
Purpose	To support the VRChat platform.
Relationship with other input/output	It is required for VRChat SDK.
Version	Unity 2019.4.30f1

[Table 13] Software Interface of VRoom Development: VRChat SDK

VRChat SDK	
Purpose	To support the development for VR and 3D world functionalities.
Relationship with other input/output	It is required for VRoom development.
Version	3

[Table 14] Software Interface of VRoom Development: VRChat Udon

VRChat Udon	
Purpose	To support the VRChat SDK as a programming language, via VRChat Udon Node Graph UI.
Relationship with other input/output	It can be used for VRoom development.
Version	Open Alpha

### 3.1.4. Communication Interfaces

VRoom uses the VRChat communication functionalities.

## 3.2. Functional Requirement

### 3.2.1. Use Case

[Table 15] Use Case of Instantiate

Use Case Name	Instantiate
Actor	Host
Description	Instantiate a new instance of the VRoom world
Normal Course	1. In world tab, the host searches the Vroom world 2. The host clicks 'New Instance' and sets proper access and region.
Precondition	VRchat must be executed
Post Condition	N/A
Assumptions	The Vroom world must be published to online

[Table 16] Use Case of Join

Use Case Name	Join
Actor	Participant
Description	Join the instance of Vroom world
Normal Course	1. In world tab, the member searches the VRoom world 2. From 'Other instances', the user clicks the instance where user want to enter
Precondition	VRchat must be executed. The instance must be created by the host user
Post Condition	N/A
Assumptions	N/A

[Table 17] Use Case of Pen &amp; Eraser

Use Case Name	Pen & Eraser
Actor	User
Description	The user can draw on the screen with pen.
Normal Course	1. The user pick up the pen. 2. The user can draw on the wall. 3. The user can erase drawing with eraser.
Precondition	N/A
Post Condition	N/A
Assumptions	N/A

[Table 18] Use Case of Projector

Use Case Name	Projector
Actor	User
Description	The user can share videos with other team members on the screen.
Normal Course	1. The user clicks 'open' from screen panel. 2. The user type a video URL. 3. The user can control video time, volume and loop. 4. The host can lock the video player so that only he can type a video URL .
Precondition	N/A
Post Condition	N/A
Assumptions	N/A

[Table 19] Use Case of Calendar

Use Case Name	Calendar
Actor	User
Description	The user can check the date on the calendar.
Normal Course	1. The user clicks the calendar. 2. The user clicks the arrow button to change a month.
Precondition	N/A
Post Condition	N/A
Assumptions	N/A

[Table 20] Use Case of Whisper zone

Use Case Name	Whisper zone
Actor	User
Description	Some users may have voice chat in certain area. And their voices can only be heard by them.
Normal Course	1. The user can communicate with voice chatting function. 2. The user can walk to whisper zone. 3. users in whisper zone can have their voice chat.
Precondition	Users can communicate with other users who are in the same meeting room.
Post Condition	N/A
Assumptions	N/A

[Table 21] Use Case of Seat

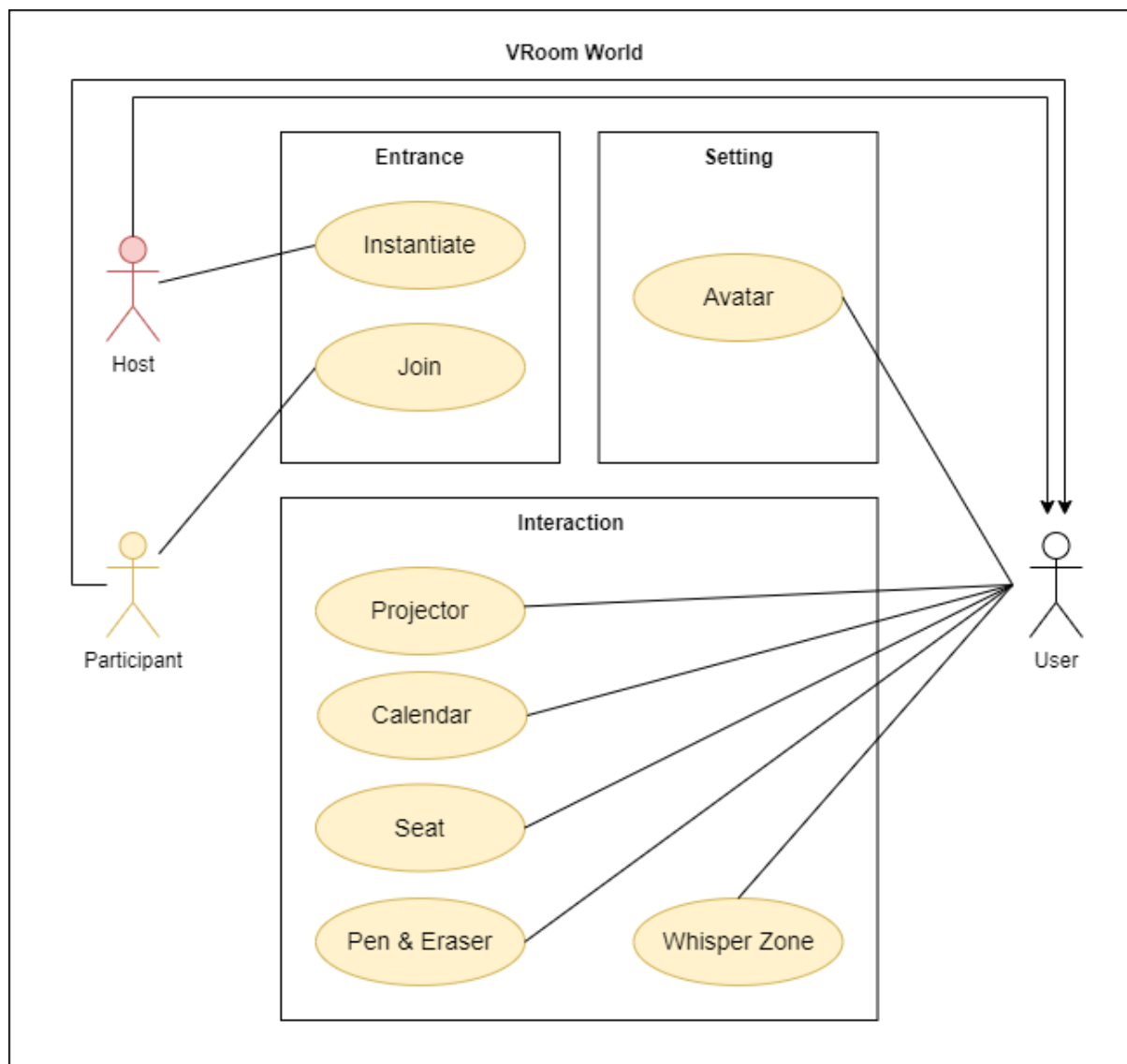
Use Case Name	Seat
Actor	User
Description	The user can sit on the chair
Normal Course	1. The user clicks the chair
Precondition	N/A
Post Condition	N/A
Assumptions	N/A

[Table 22] Use Case of Avatar

Use Case Name	Avatar
Actor	User
Description	The user can select avatar.

Normal Course	1. When the user enters the instance, some avatars are shown. 2. The user can choose one of them or instead use your own avatar.
Precondition	N/A
Post Condition	The user will be transformed into an avatar of choice.
Assumptions	N/A

### 3.2.2. Use Case Diagram



[Figure 1] Use Case Diagram

### 3.3. Performance Requirements

This section describes the performance requirements of VRoom. This section is divided into two parts: Static numerical requirements and dynamic numerical requirements. Static numerical requirements cover items that are not related to time. On the other hand, dynamic numerical requirements cover items that are related to time. The following requirements are based on estimates and may be changed in the final application.

### **3.3.1. Static Numerical Requirements**

- The system management tool supports only one administrator
- The system supports only one simultaneous user for each device. The system does not support multiple connections on the same device. However, after disconnecting, user can switch accounts and access
- The system should run smoothly on devices with at least 4GB RAM and Intel® i5-4590 / AMD FX 8350 CPU. It requires a 64-bit processor and operating system, Windows 8.1 and Windows 10

### **3.3.2. Dynamic Numerical Requirements**

- The system runs smoothly for at least 100 simultaneous users. And the system can handle at least 1,000 active user accounts and profiles
- Each account must be activated within 10 seconds of linking
- The login process should be completed within 10 seconds
- Entering the meeting room world should be completed within 30 seconds

## **3.4. Design Constraints**

The system should contain only components that can be distributed under the MIT license. The system must be designed to run on VRChat.

## **3.5. Standards Compliance**

All programs in the system follow conventional programming techniques. We will use VRChat Udon which is a programming language built by the VRChat development team and follow their guidelines.

### **3.5.1. Compliance for Submitting World**

- Submit one world per user per seven days to Community Labs
- Update your world as often as you like

- If your world is already Public, you don't need to re-submit the world if you update it
- If your world or any content in the world violates the VRChat Terms of Service or the Community Guidelines, your ability to submit worlds to Community Labs will be suspended for a period of time
- Try to keep your worlds under 200MB
- Test your lighting

### **3.5.2. Compliance for Performance**

- Aim for greater than 45 FPS with a single VR user at the spawn
- Do not use shaders that are not VR-compatible
- Be very careful with post-processing effects
- Do not put more than 2 video players in a room
- Do not put more than 1 mirror in your room and make any and all mirrors in your world toggleable
- Do not overuse real time lights

### **3.5.3. Compliance for General**

- Do not use Blendfiles for import
- Test your world and test your world in VR
- Test your lighting

## **3.6. Software System Characteristics**

Software system characteristics are well understood through non-functional requirements. This section shows several non-functional requirements of the system. There are product requirements, organization requirements, and external requirements as follows.

### **3.6.1. Product Requirements**

Product Requirements describes the requirements that should be satisfied when the software is operating. Our system should satisfy following requirements.

#### **3.6.1.1. Usability Requirements**

Usability is one of the most important non-functional requirements of our system. The system should be easy to be used and should be organized in a way that user errors are



minimized. Each user shall be able to use all the system functions instinctively. This means that user interface should be simple, instinctive, and easy to use.

#### **3.6.1.2. Performance Requirements**

The performance of a system depends on how quickly the system respond to user requests. For requests such as entering the meeting room, the system must be able to provide it up to 30 seconds.

#### **3.6.1.3. Reliability Requirements**

The system should provide its service at any given time. The meeting room should be able to be used by users at any time. Also, only allowed users must be able to enter the meeting room.

#### **3.6.1.4. Security Requirements**

The users should be properly authenticated before using the system. It should make sure that an unauthorized user cannot access the system. Also, only administrator can access every user's confidential information such as user's personal Information, ID, and password.

### **3.6.2. Organizational Requirements**

These requirements are derived from policies and procedures in the customer's and developer's organizations.

#### **3.6.2.1. Operational Requirements**

Users of the system shall identify themselves using their Steam account. The system provides a metaverse meeting room to people who usually don't use it. Thus, it should be user-friendly interface. User would enter the meeting room within 1 minutes. The system works at any time when users need the meeting room.

### **3.6.3. External Requirements**

It describes all requirements that are derived from external environment of the system.

#### **3.6.3.1. Safety / Security Requirements**

The system should guarantee that the external system cannot access personal information. It must ensure to prevent user's data without being lost.

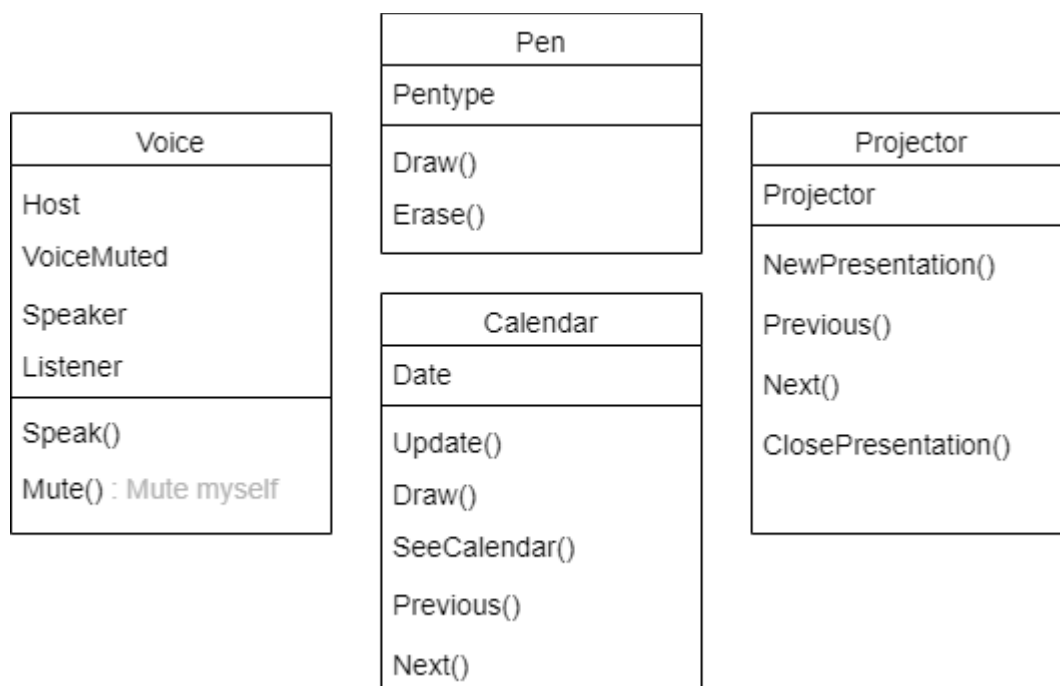
### 3.6.3.2. Regulatory Requirements

The software must not infringe on open source licenses when using open source software. Also, the system should not violate user's privacy according to national privacy standard.

## 3.7. Organizing the Specific Requirements

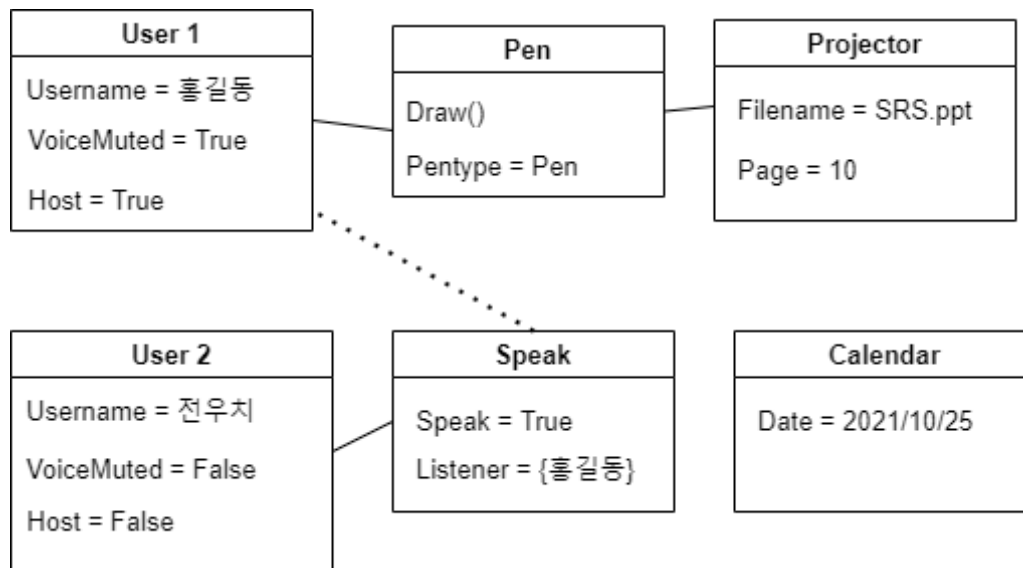
Different classes of systems need different organizations of requirements. There is no one optimal organization for all systems. We've organized the requirements by functional hierarchy. In this section, we'll use Unified Modeling Language (UML) to show the relationships between the system components, the system and its environment.

### 3.7.1. Class Model



[Figure 2] Class Model

### 3.7.2. Object Model



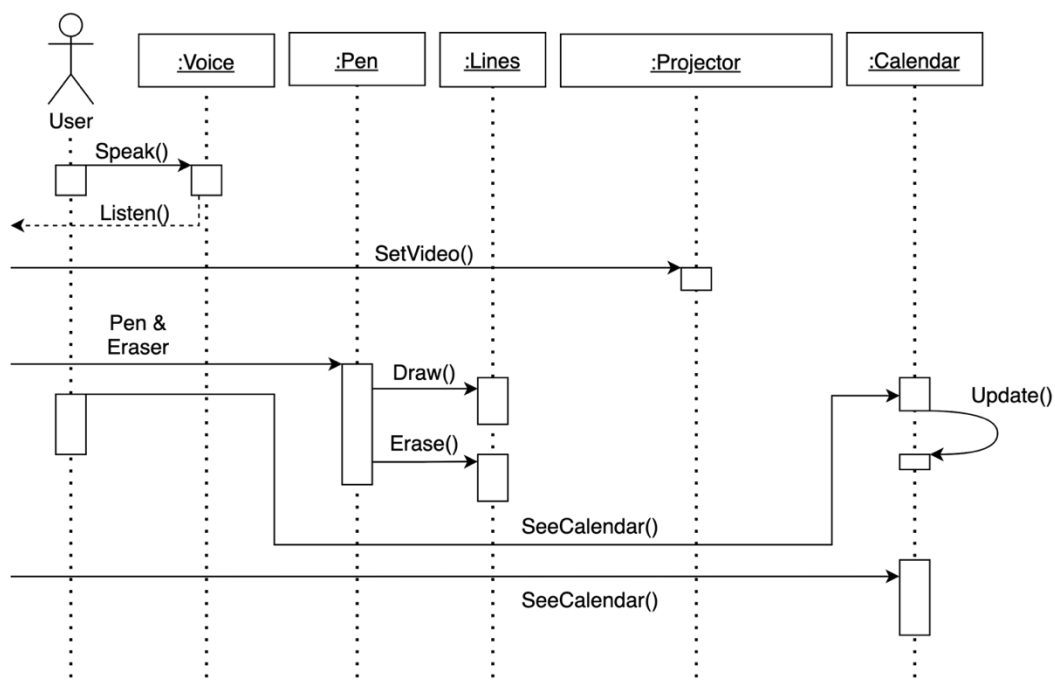
[Figure 3] Object Model

### 3.7.3. Interaction Model

See 3.2.2 Use Case diagram.

### 3.7.4. Behavior Model

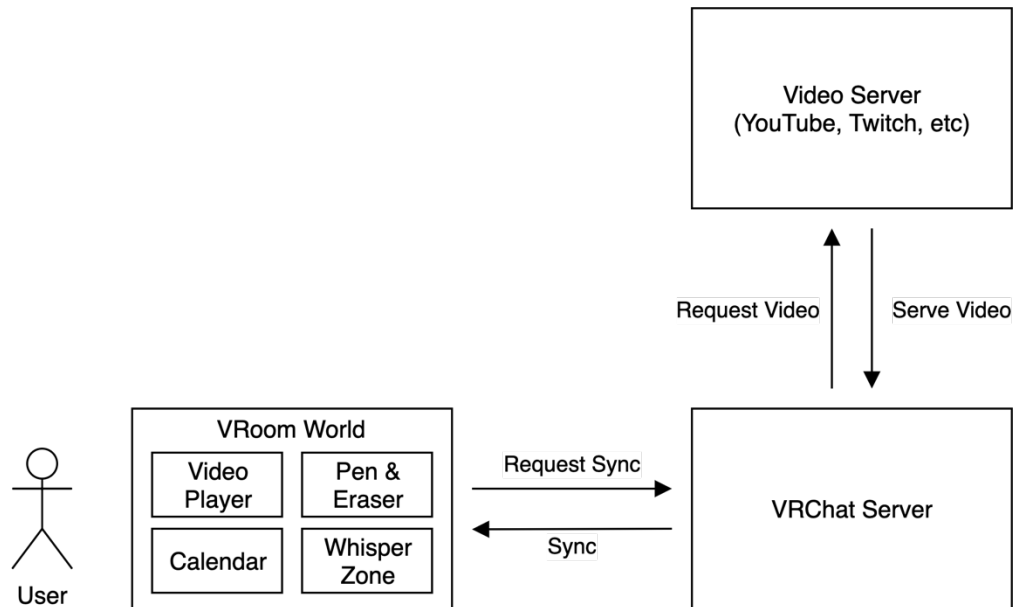
#### 3.7.4.1. Sequence Diagram



[Figure 4] Sequence Diagram

### 3.8. System Architecture

In this section, we present a high-level overview of the anticipated system architecture, showing the distribution of functions across system modules. We use client-server pattern.



[Figure 5] System Architecture

### 3.9. System Evolution

In this section, we'll describe the fundamental assumptions on which the system is based and any anticipated changes due to hardware evolution, changing user needs, and etc. This section is useful for system designers as it may help them avoid design decisions that would constrain possible changes to the system.

#### 3.9.1. Limitation and Assumption

VRoom is only available in VRChat environments. In other words, all users must install Steam and VRChat. This makes communication only possible by voice chat, and typing is impossible. VRChat assumes that users use VR devices, however, ordinary users rarely have them. Thus, we assume that VRoom users only use desktop or laptop and have microphone and speaker devices.

#### 3.9.2. Evolutions of Hardware and Change of User Requirements

Recently, with the growing interest in metaverse, many people predicted that the VR market would grow together. In fact, however, the growth of VR is relatively slow compared to the interest in VR. It's because VR hardware has not yet developed significantly. If these devices improve further and if many people use them, VRoom will also need to evolve accordingly.

Since we didn't develop the motion-capturing actions that cannot be used in desktop or laptop, features corresponding to the body movement of the user should be added. For example, generate pens by waving your hand or call pamphlets by knocking the table.

## 4. Supporting Information

### 4.1. Document History

[Table 23] Document History

Date	Version	Description	Writer
2021. 10. 27.	1.0	Addition of 1 and 4	Kim Hongbeen
2021. 10. 27.	1.1	Addition of 2.1, 2.2	Lee Jiwoo
2021. 10. 27.	1.2	Addition of 2.3, 2.4, 2.5	Kim Minhee
2021. 10. 27.	1.3	Addition of 3.1	Kim Jinwoong
2021. 10. 27.	1.4	Addition of 3.2	Hong Seongjun
2021. 10. 27.	1.5	Addition of 3.3, 3.4, 3.5, 3.6	Seo Chaeyeon
2021. 10. 27.	1.6	Addition of 3.7, 3.8, 3.9	Lee Hyewon
2021. 10. 29.	2.0	Unification of terms	Kim Hongbeen
2021. 10. 29.	2.1	Revision of 2.3, 2.4	Kim Minhee
2021. 10. 29.	2.2	Revision of 3.7	Lee Hyewon
2021. 10. 29.	2.3	Revision of 3.2	Hong Seongjun
2021. 10. 29.	2.4	Revision of 3.1	Kim Jinwoong
2021. 10. 29.	2.5	Revision of 3.4	Seo Chaeyeon
2021. 10. 30.	2.6	Revision of 1	Kim Hongbeen
2021. 10. 31.	2.7	Revision of 3.5	Seo Chaeyeon
2021. 10. 31.	2.8	Final	Kim Jinwoong
2021. 11. 20	3.1	Revision of 1	Kim Hongbeen
2021. 11. 20	3.2	Revision of 2.1, 2.2	Lee Jiwoo
2021. 11. 20	3.3	Revision of 3.2	Hong Seongjun