



VRoom

Virtual Meeting Room Service

Software Design Specification

2021.11.21.

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

1. PREFACE	7
1.1. Readership	7
1.2. Scope	7
1.3. Objective.....	7
1.4. Document Structure.....	7
2. INTRODUCTION	9
2.1. Objectives.....	9
2.2. Applied Diagrams	9
2.2.1. UML.....	9
2.2.2. Use case Diagram.....	9
2.2.3. Sequence Diagram	10
2.2.4. Class Diagram.....	10
2.2.5. Context Diagram	10
2.2.6. System Architecture Diagram.....	11
2.3. Applied Tools	11
2.3.1. Drawio	11
2.4. Project Scope	11
2.5. References	11
3. SYSTEM ARCHITECTURE: OVERALL	13
3.1. System Organization	13
3.1.1. Context Diagram	14
3.1.2. Sequence Diagram	14
3.1.3. Use Case Diagram.....	15
4. SYSTEM ARCHITECTURE: FRONTEND.....	15
4.1. Objectives.....	15
4.2. Subcomponents.....	15
4.2.1. Projector.....	15
4.2.1.1. Attributes	16
4.2.1.2. Methods	16
4.2.1.3. Class Diagram	17
4.2.1.4. Sequence Diagram	18
4.2.2. Whisper Zone	20
4.2.2.1. Attributes	21

4.2.2.2. Methods	21
4.2.2.3. Class Diagram	22
4.2.2.4. Sequence Diagram	23
4.2.3. Calendar.....	24
4.2.3.1. Attributes	24
4.2.3.2. Methods	24
4.2.3.3. Class Diagram	25
4.2.3.4. Sequence Diagram	26
4.2.4. Pen.....	26
4.2.4.1. Attributes	26
4.2.4.2. Methods	27
4.2.4.3. Class Diagram	28
4.2.4.4. Sequence Diagram	29
5. TESTING PLAN	30
5.1. Objectives.....	30
5.2. Testing Policy.....	30
5.2.1. Development Testing.....	30
5.2.2. Release Testing.....	30
5.2.3. User Testing	30
5.3. Test Case.....	31
5.3.1. Pen.....	31
5.3.2. Projector.....	31
5.3.3. Calendar.....	31
5.3.4. Whisper Zone.....	31
6. DEVELOPMENT PLAN.....	32
6.1. Objectives.....	32
6.2. Frontend Environment.....	32
6.2.1. Autodesk Inventor	32
6.2.2. Unity	32
6.2.3. VRChat.....	32
6.3. Backend Environment	33
6.3.1. GitHub	33
6.3.2. VRchat	33
6.4. Open source	34
6.4.1. QvPen	34

6.4.2. BetterAudio	34
6.4.3. UdonSharpVideo	34
6.4.4. UdonSharp	35
6.5. Constraints	35
6.6. Assumptions and Dependencies	35
7. SUPPORTING INFORMATION	36
7.1. Software Design Specification	36
7.2. Document History	36

LIST OF FIGURES

[Figure 1] Overall System Architecture	13
[Figure 2] Overall Context Diagram	14
[Figure 3] Overall Sequence Diagram	14
[Figure 4] Use Case Diagram	15
[Figure 5] Class diagram – projector.....	17
[Figure 6] Sequence diagram – Information	18
[Figure 7] Sequence diagram – Volume	19
[Figure 8] Sequence diagram – VideoController.....	20
[Figure 9] Class diagram – Whisper zone.....	22
[Figure 10] Sequence diagram – Information	23
[Figure 11] Sequence diagram – Volume	24
[Figure 12] Class diagram – Calendar	25
[Figure 13] Sequence diagram – Button	26
[Figure 14] Class diagram – Pen	28
[Figure 15] Sequence diagram – Pen	29
[Figure 16] Autodesk Inventor logo	32
[Figure 17] Unity logo.....	32
[Figure 18] VRChat logo	33
[Figure 19] GitHub logo.....	33
[Figure 20] VRChat logo	33
[Figure 21] UdonSharpVideo image.....	34

LIST OF TABLES

[Table 1] Document History	36
----------------------------------	----

1. Preface

1단원 Preface는 본 문서의 독자층, 범위, 목표하는 바, 문서의 구조에 대한 내용을 담고 있습니다.

1.1. Readership

본 software design 문서는 7개의 단원으로 구성되어 있고, 각 단원마다 세부단원들이 존재합니다. 문서의 구조는 위의 목차와 일치합니다. 본 문서의 작성자는 Team 2입니다. 그리고 2021년 2학기 소프트웨어공학개론 수업의 이은석 교수님과 조교님, Team 2 소속 학생들이 해당 문서의 주요 독자입니다.

1.2. Scope

해당 design specification은 메타버스 환경 내에서 이용할 수 있는 미팅룸 서비스를 구현하기 위해 소프트웨어공학의 단계로써 사용될 것입니다.

1.3. Objective

본 software design specification 문서의 주요 목적은 team 2의 메타버스 미팅룸 서비스인 VRoom을 구현하기 위해 필요한 기술적인 내용을 제공하는 것입니다. 따라서 본 문서는 VRoom을 구현하기 위한 system architecture와 design에 대한 결정사항들을 설명합니다. 더 나아가 요구사항 명세서에 명시되었던 내용 중 사용 class의 역할 및 기능에 대해서도 구체화합니다. Use case diagram, class diagram, sequence diagram, context diagram 등을 사용하여 표현하였습니다. 마지막으로 시스템 개발 및 테스트 계획도 내보입니다. 그렇기에 본 문서의 독자는 VRoom서비스의 stakeholder들과 개발자들, 디자이너들, 테스트하는 사람들이라고 예측됩니다.

1.4. Document Structure

- 1. Preface: 본 문서의 독자층과 범위와 구조, 그리고 시스템이 목표하는 바를 설명합니다.
- 2. Introduction: design 단계에서 본 프로젝트에 적용한 다양한 도구와 다이어그램을 설명합니다.
- 3. System Architecture: Overall: 본 시스템의 전체적인 구조에 대해 설명합니다.
- 4. System Architecture: Frontend: 본 시스템에 사용된 subcomponent들에 대해 설명합니다.

- 5. Testing plan: 본 시스템이 잘 실행되는지 test할 계획 및 방법에 대해 설명합니다.
- 6. Development plan: 프로그램 개발을 위해 사용한 개발 도구와 환경을 설명합니다.
- 7. Supporting Information: 본 문서의 software design specification방식과 document history를 담았습니다.

2. Introduction

이 프로젝트는 코로나로 인해 어려워진 사람 간의 미팅과 오프라인 장소 대여를 대체할 수 있을 metaverse meeting room인 Vroom을 개발하기 위한 프로젝트입니다. 이 meeting room은 오프라인 스터디룸의 형태와 유사하게 만들어졌지만 시간과 공간의 제약 없이 온라인에서 미팅을 할 수 있도록 합니다. 이 Design Specification Document는 project에서 구현될 design을 묘사합니다. 이 문서의 design들은 이전 문서인 Requirement Specification의 requirement들을 따릅니다.

2.1. Objectives

이 장에서는 design단계에서 본 프로젝트에 적용한 다양한 도구와 다이어그램을 설명합니다.

2.2. Applied Diagrams

2.2.1. UML

UML은 Unified Modeling Language의 약자로서, 소프트웨어를 모델링하고 문서화하는데 이용되는 정형화된 접근법입니다. 매우 자주 쓰이는 비즈니스 프로세스 모델링 기법으로 비즈니스 분석가, 소프트웨어 설계자, 그리고 개발자들이 주로 이용합니다. 소프트웨어 시스템의 비즈니스 프로세스, 구조와 동작을 기술, 명시, 설계, 문서화하는 데 사용됩니다. 이 때 소프트웨어 시스템은 새로 만들어지는 것, 원래 존재하던 것 둘 다를 포함합니다. UML은 다양한 애플리케이션 도메인(예: 은행, 금융, 인터넷, 항공우주, 의료 등)에 적용할 수 있습니다. 모든 주요 object, 컴포넌트 소프트웨어 개발 방법, 다양한 구현 플랫폼(예: J2EE, .NET) 등과 함께 사용할 수 있습니다. 소프트웨어 구성 요소의 다이어그램 위주의 표현을 기반으로 합니다. 시각적 표현을 사용함으로써 소프트웨어 또는 비즈니스 프로세스에서 발생할 수 있는 결함이나 오류를 더 잘 이해할 수 있습니다.

2.2.2. Use case Diagram

시스템이 충족해야 할 기능적 요구사항(Functional Requirement)은 시스템의 주춧돌과도 같은 부분입니다. Use case Diagram은 시스템의 high-level 요구사항을 분석하는 데 사용됩니다. 이러한 요구사항은 다양한 사용 사례(Use case)를 통해 표현됩니다. 이 UML 다이어그램의 세 가지 주요 구성 요소는 다음과 같습니다: 기능적 요구사항 - 사용 사례, 동작을 설명하는 동사로 표현됩니다. 사용자(User) - 사용자는 시스템과의 상호작용을 하며, 사용자는 인간, 조직 또는 내부 또는 외부 응용 프로그램이 될 수

있습니다. 사용자와 사용 사례(Use case) 간의 관계 - 직선 화살표를 사용하여 표현됩니다.

2.2.3. Sequence Diagram

Sequence Diagram은 컴퓨터 과학 분야뿐만 아니라 비즈니스 프로그램 개발을 위한 설계 단계의 모델로서도 가장 중요한 UML 다이어그램이라고 할 수 있습니다. 최근에는 시각적으로 자명하고 이해가 쉽다는 특성 때문에 비즈니스 프로세스를 묘사하는 데 자주 쓰입니다. 이름에서 알 수 있듯이, 시퀀스 다이어그램은 User와 Object 사이에서 일어나는 메시지와 상호작용의 순서를 설명합니다. User나 Object는 필요할 때나 다른 객체가 그들과 통신하기를 원할 때에만 활성화될 수 있습니다. 모든 의사소통은 시간 순서에 따라 표현됩니다.

2.2.4. Class Diagram

클래스 다이어그램은 시스템의 클래스, 속성, 기능(또는 메소드), 그리고 객체(Object) 간의 관계를 보여줌으로써 시스템의 구조를 설명하는 정적 구조 다이어그램의 일종입니다. 다이어그램의 다양한 구성 요소는 실제로 프로그래밍될 클래스, 주요 객체, 또는 클래스와 객체 간의 상호 작용 등 다양한 요소들을 나타낼 수 있습니다. 하나의 클래스는 세 부분으로 나뉘어 직사각형으로 표현되는데, 맨 위에는 클래스의 이름, 가운데는 클래스의 attribute, 맨 밑에는 클래스에서 사용될 수 있는 function(또는 method)으로 구성됩니다. 클래스와 하위 클래스는 각 객체 간의 정적 관계를 표시하기 위해 함께 그룹화됩니다.

2.2.5. Context Diagram

System Context Diagram(레벨 0 DFD라고도 함)은 데이터 흐름도에서 가장 높은 수준이며, 전체 시스템을 나타내는 프로세스 하나만 포함합니다. 이는 모델링할 시스템의 배경과 경계를 설정합니다. 시스템과 외부의 행위자(Users) 사이의 정보 흐름을 식별합니다. Context Diagram은 일반적으로 요구사항 문서에 포함되어 있습니다. 모든 프로젝트 이해 당사자가 읽어야 하므로 그들이 항목을 이해할 수 있도록 쉬운 언어로 작성해야 합니다. 이 다이어그램의 목적은 전체 시스템 요구사항 및 제약사항들을 개발할 때 고려해야 하는 외부 요인 및 이벤트에 초점을 맞추는 것입니다. 시스템 컨텍스트 다이어그램은 프로젝트 초기에 조사 대상 범위를 결정하기 위해 종종 사용됩니다. 따라서, 문서 내에서, 시스템 컨텍스트 다이어그램은 시스템과 상호 작용할 수 있는 모든 외부 요소(entity)를 나타냅니다. 전체 소프트웨어 시스템은 단일 프로세스로 표시됩니다. 이러한 다이어그램은 모든 외부 엔티티, 상호 작용하는 시스템, 그리고 환경에 둘러싸인 내부 구조 등 세부 사항 없이 시스템의 중점 부분만을 그립니다.

2.2.6. System Architecture Diagram

아키텍처 다이어그램은 소프트웨어 시스템의 전체적인 개요와 구성 요소 간의 관계, 제약, 경계를 추상화하는 데 사용되는 시스템 다이어그램입니다. 소프트웨어 시스템의 물리적 배치와 진화 로드맵에 대한 전반적인 뷰를 제공하기 때문에 중요한 diagram으로 평가됩니다.

2.3. Applied Tools

2.3.1. Drawio

Google drive에 설치하여 사용할 수 있는 프로그램으로, 다양한 종류의 diagram을 지원하며 사용자가 자유롭게 object를 배치할 수 있다는 장점이 있습니다. 이 문서에서는 UML로 diagram을 그리는 데에 사용되었습니다.

2.4. Project Scope

VRoom은 VRChat 플랫폼에서 제공되는 가상 미팅 룸 서비스입니다. VRoom은 그룹 과제 회의, 그룹 스터디, 비즈니스 프레젠테이션 등 모든 종류의 미팅에 사용할 수 있습니다. 미팅 참가자의 비디오를 스트리밍하지 않기 때문에, 줌 피로 - 영상통화 중 참가자들이 다른 참가자의 화면에 어떻게 나타날지 끊임없이 의식하며 스트레스를 받는 것 - 문제에서 자유롭습니다. 또한 3D 환경에서 구현되기 때문에 기존 영상통화 서비스보다 뛰어난 몰입감과 사실감을 제공합니다. 따라서 우리는 Vroom이 코로나 대유행으로 촉발된 재택근무 문화를 더욱 발전시킬 것으로 기대됩니다.

2.5. References

이 Design Specification 문서의 사용자는 아래의 문서를 참조할 수 있습니다:

- MerlinVR, Github, UdonSharp
“<https://github.com/MerlinVR/UdonSharp/wiki/udonsharp>” (accessed 2021.11.21)
- MerlinVR, Github, UdonSharpVideo “<https://github.com/MerlinVR/USharpVideo/wiki>” (accessed 2021.11.21)
- VRChat Guides, Player Audio, “<https://docs.vrchat.com/docs/player-audio>” (accessed 2021.11.21)
- Unity Documentation, Unity User Manual 2020.3 (LTS)
“<https://docs.unity3d.com/Manual/index.html>” (accessed 2021.11.21)
- Autodesk, Inventor, “<https://www.autodesk.co.kr/products/inventor/overview>” (accessed 2021.11.21)

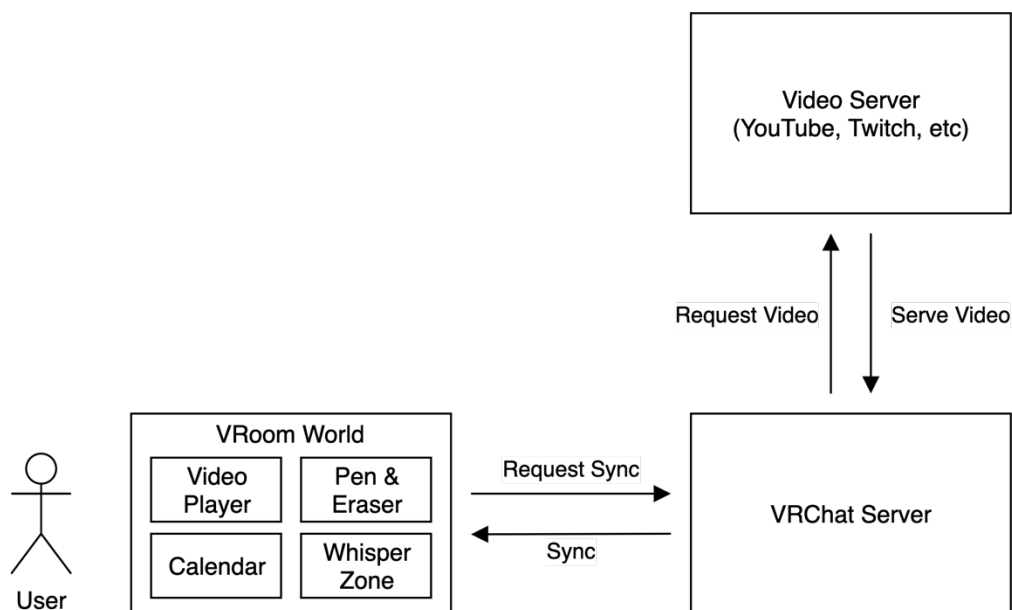
- Unity, Unity kr, "<https://unity.com/kr>" (accessed 2021.11.21)
- VRChat, hello.vrchat, "<https://hello.vrchat.com>" (accessed 2021.11.21)

3. System Architecture: Overall

3단원 System Architecture에서는 시스템의 전체적인 구조에 대해 설명합니다.

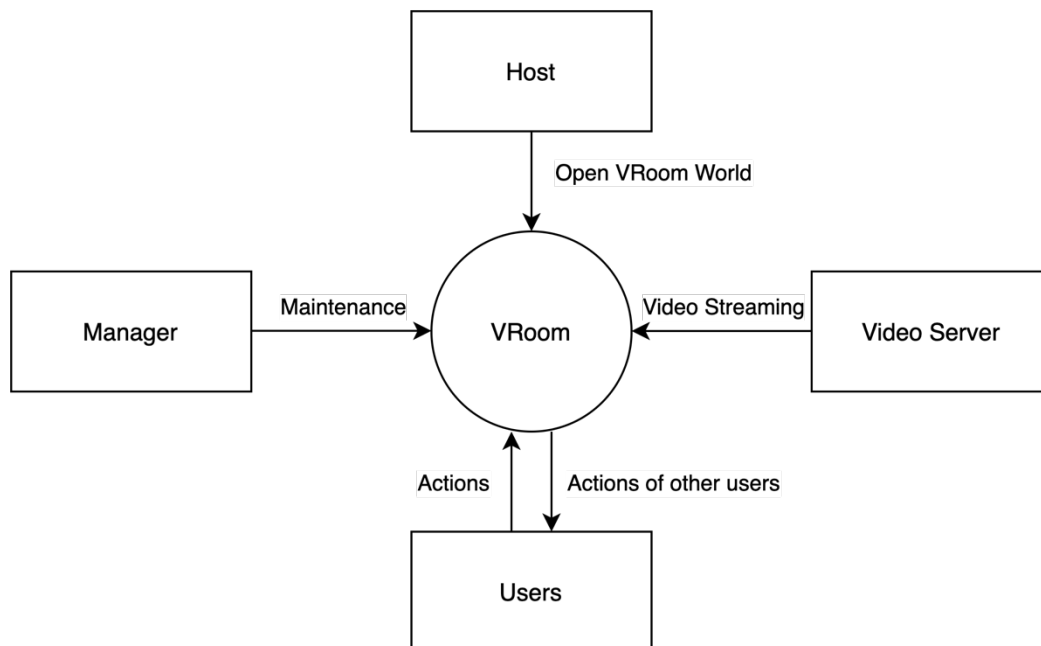
3.1. System Organization

VRoom은 VRChat을 기반으로 하여 프론트엔드 위주로 설계되었습니다. 사용자는 VR기기를 이용해 Vroom World에 접속할 수 있습니다. 프론트엔드 부분에서는 사용자와의 상호작용을 담당합니다. 프론트엔드의 기능으로는 Video Player, Pen & Eraser, Calendar, Whisper zone이 존재합니다. Vroom World와 VRChat 서버는 Vroom World에서 Sync를 요청하면 VRChat 서버에서 Sync를 함으로써 통신합니다. Video Player에서는 사용자에게서 입력 받은 동영상 링크를 VRChat 서버에 전송하는데, 이때 VRChat 서버에서는 YouTube, Twitch같은 요청된 동영상의 서버와 통신하여 동영상을 전달받습니다. 이후 VRoom World와 Sync할 때 Vroom World에 동영상을 제공합니다.



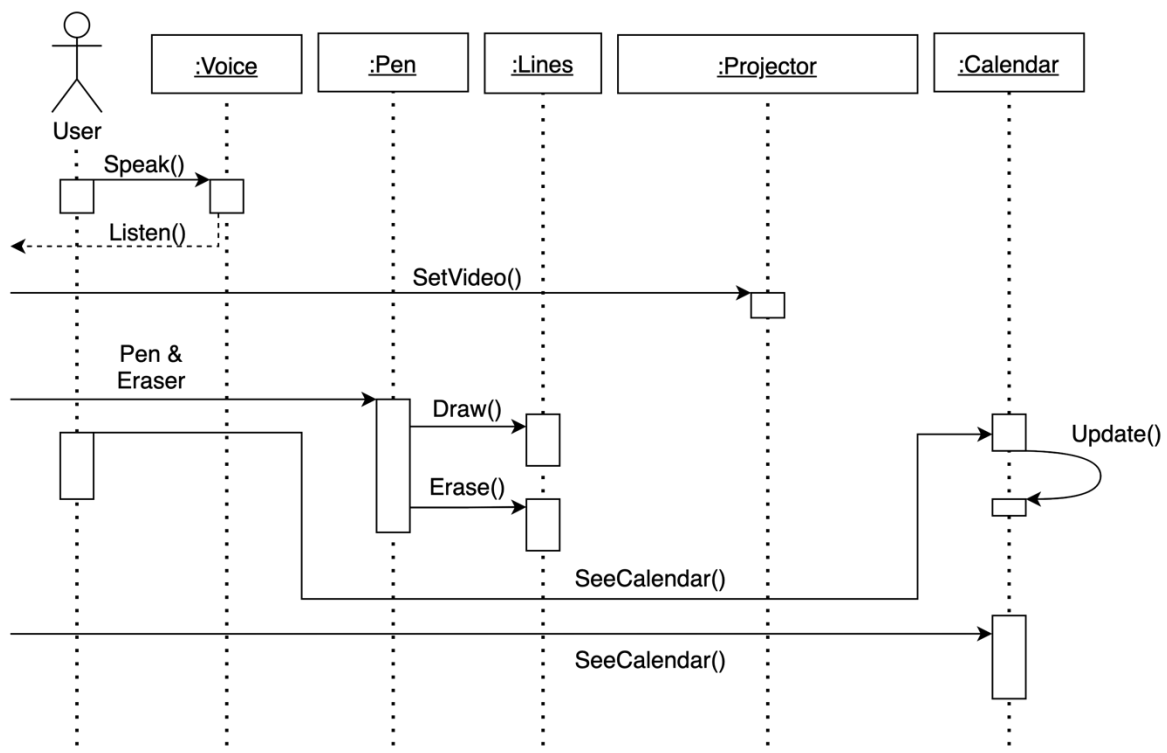
[Figure 1] Overall System Architecture

3.1.1. Context Diagram



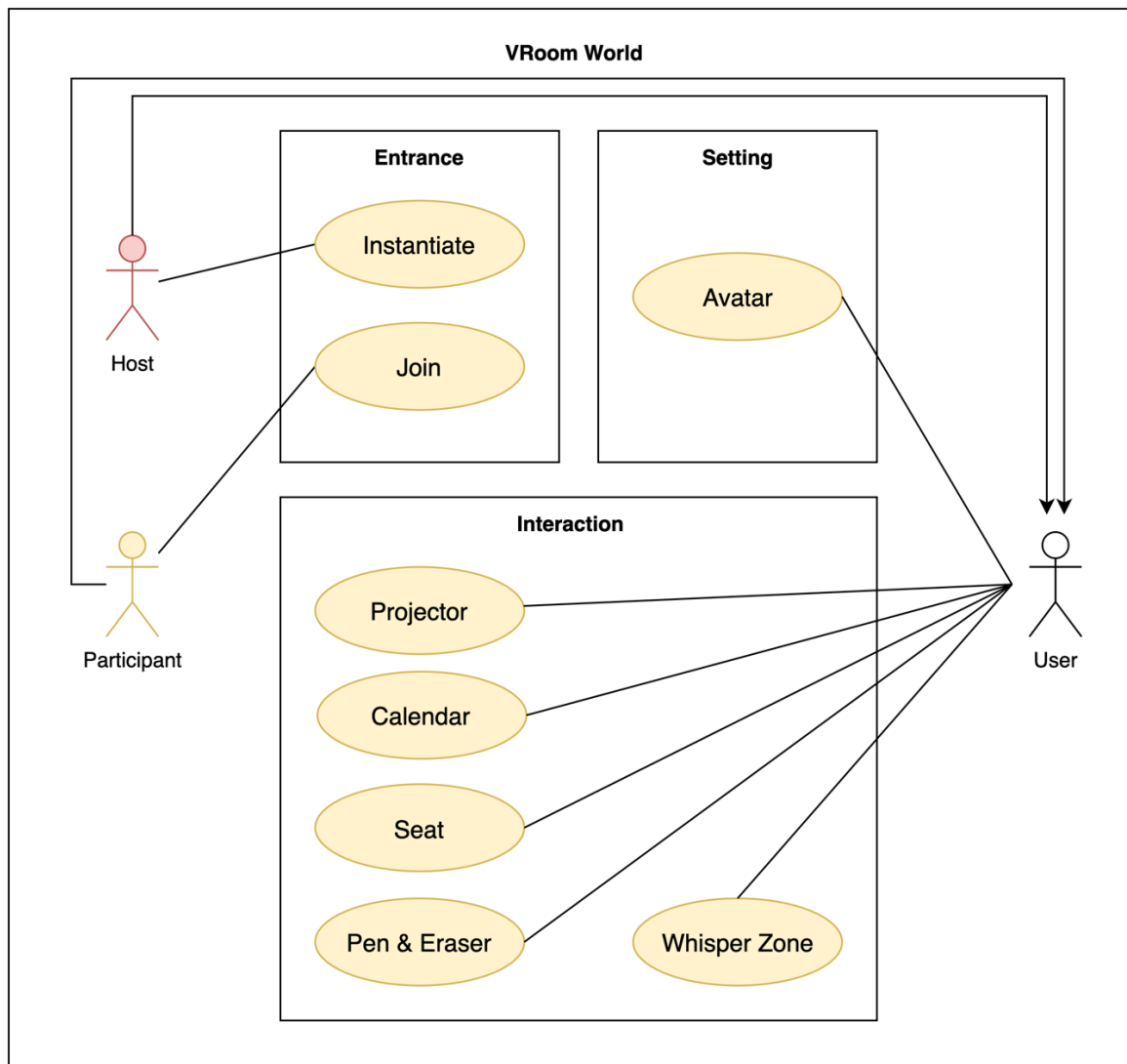
[Figure 2] Overall Context Diagram

3.1.2. Sequence Diagram



[Figure 3] Overall Sequence Diagram

3.1.3. Use Case Diagram



[Figure 4] Use Case Diagram

4. System Architecture: Frontend

4.1. Objectives

본 시스템에 사용된 subcomponent들에 대해 설명합니다.

4.2. Subcomponents

4.2.1. Projector

사용자들은 비디오 플레이어를 통해 일반 비디오나 스트리밍 비디오를 시청할 수 있습니다. 방 안의 모든 미팅 참여자들은 동일한 비디오를 시청합니다.

비디오의 URL 링크를 입력하면 스크린 화면에 해당 비디오가 불러와집니다. 스트리밍 중인 비디오도 불러오는 것이 가능합니다. 비디오에 대해 현재 보여주고 있는 시간, 분, 초의 값과 소리 크기, 반복재생할지의 여부를 조절할 수 있습니다. 또한 비디오를 다시 불러올 수도 있습니다. 또한 미팅 호스트가 URL을 자신만 입력하도록 잠금설정을 할 수도 있습니다.

4.2.1.1. Attributes

비디오 플레이어와 관련된 속성들이 존재합니다.

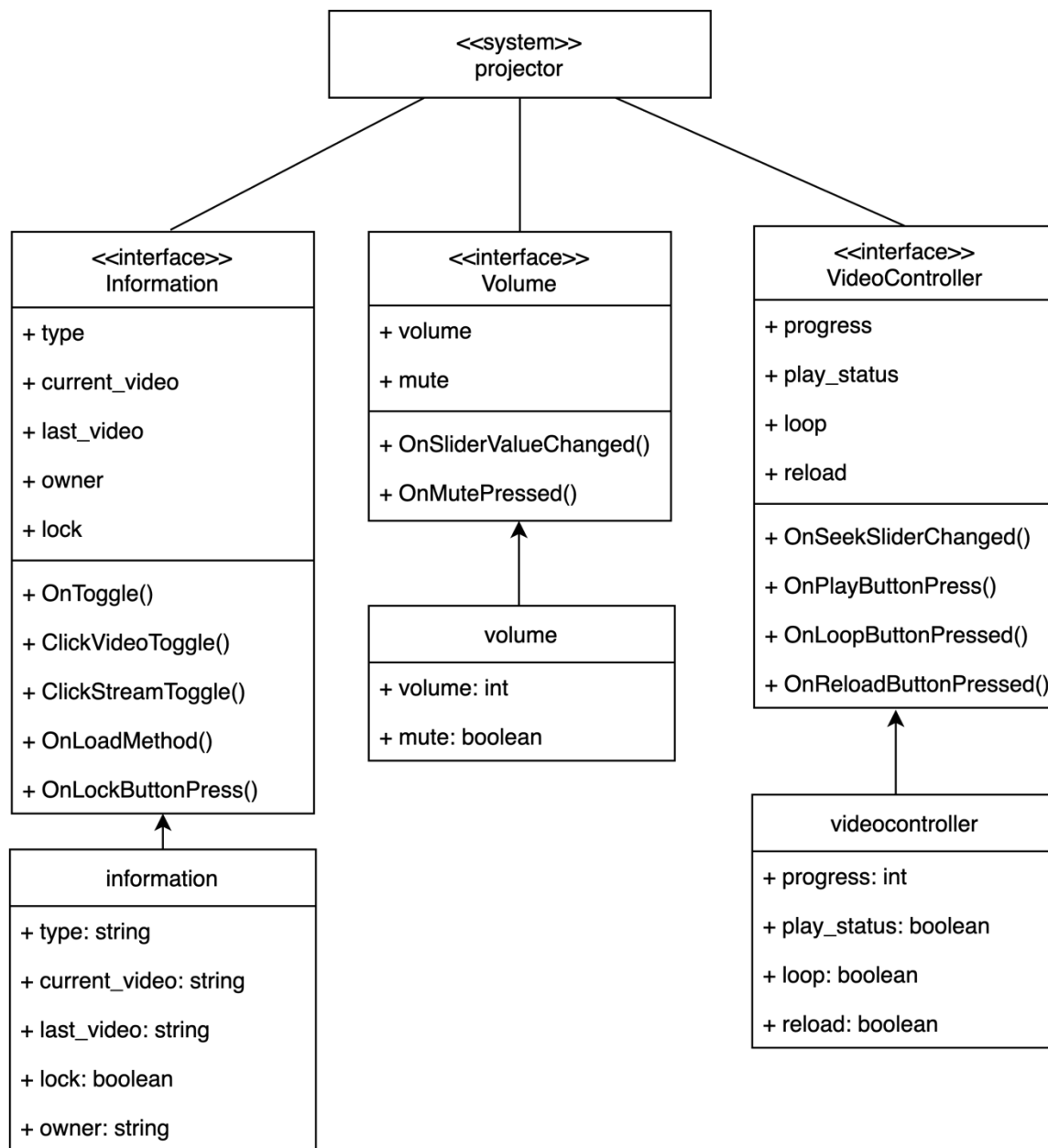
- **type**: Video or Stream, 즉 video URL이 일반 video URL인지 스트리밍 video URL인지를 의미합니다.
- **current_video**: 현재 상영중인 video의 URL입니다.
- **last_video**: 마지막으로 상영했던 video의 URL입니다.
- **progress**: 비디오가 현재 보여주고 있는 시간, 분, 초의 값입니다.
- **play_status**: 비디오가 현재 재생중인지 멈춤상태인지를 의미합니다.
- **owner**: 현재 상영중인 video의 URL 링크를 입력한 사용자입니다.
- **volume**: 비디오의 소리 크기입니다.
- **mute**: 비디오의 소리가 음소거되었는지 아닌지를 의미합니다.
- **lock**: 미팅룸의 호스트만 video URL 링크를 입력할 수 있도록 잠겨있는지를 나타냅니다. 만약 잠겨있다면 미팅 호스트만 링크를 입력할 수 있습니다. 잠겨있지 않다면 모든 미팅 참여자가 링크를 입력할 수 있습니다.
- **loop**: 현재 상영중인 비디오를 반복재생할지를 의미합니다.
- **reload**: 비디오를 다시 load할지를 의미합니다.

4.2.1.2. Methods

비디오 플레이어와 관련된 method들은 다음과 같습니다.

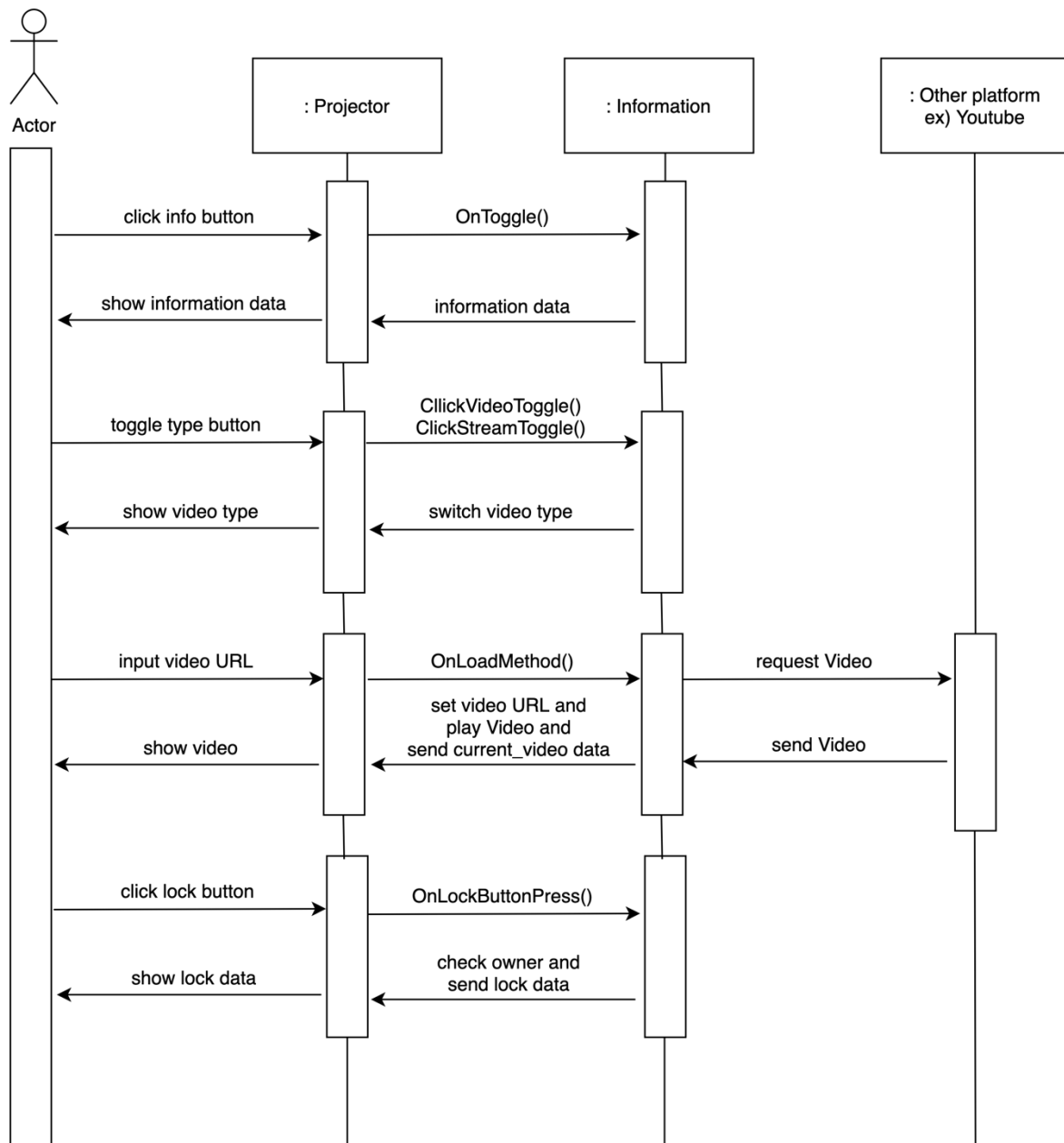
- **OnToggle()**: info버튼 클릭
- **OnReloadButtonPressed()**: reload 버튼 클릭
- **OnLoopButtonPressed()**: loop 버튼 클릭
- **ClickVideoToggle()**: video toggle 클릭
- **ClickStreamToggle()**: stream toggle 클릭
- **OnSeekSliderChanged()**: 비디오 재생 슬라이더 조정
- **OnPlayButtonPress()**: 비디오 재생/멈춤 버튼 클릭
- **OnSliderValueChanged()**: 볼륨 크기 슬라이더 조정
- **OnMutePressed()**: 음소거 버튼 클릭
- **OnLockButtonPress()**: URL 입력 잠금 버튼 클릭
- **OnLoadMethod()**: URL 입력

4.2.1.3. Class Diagram

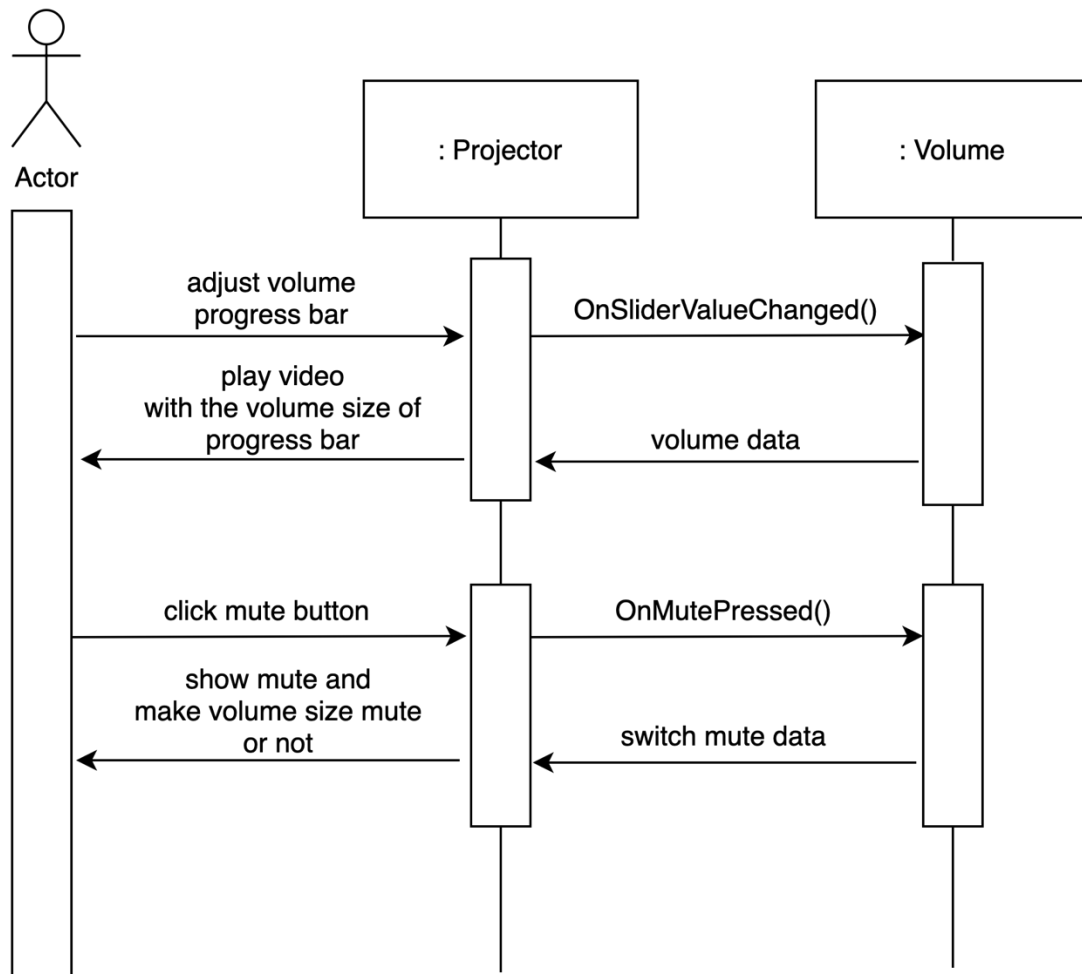


[Figure 5] Class diagram – projector

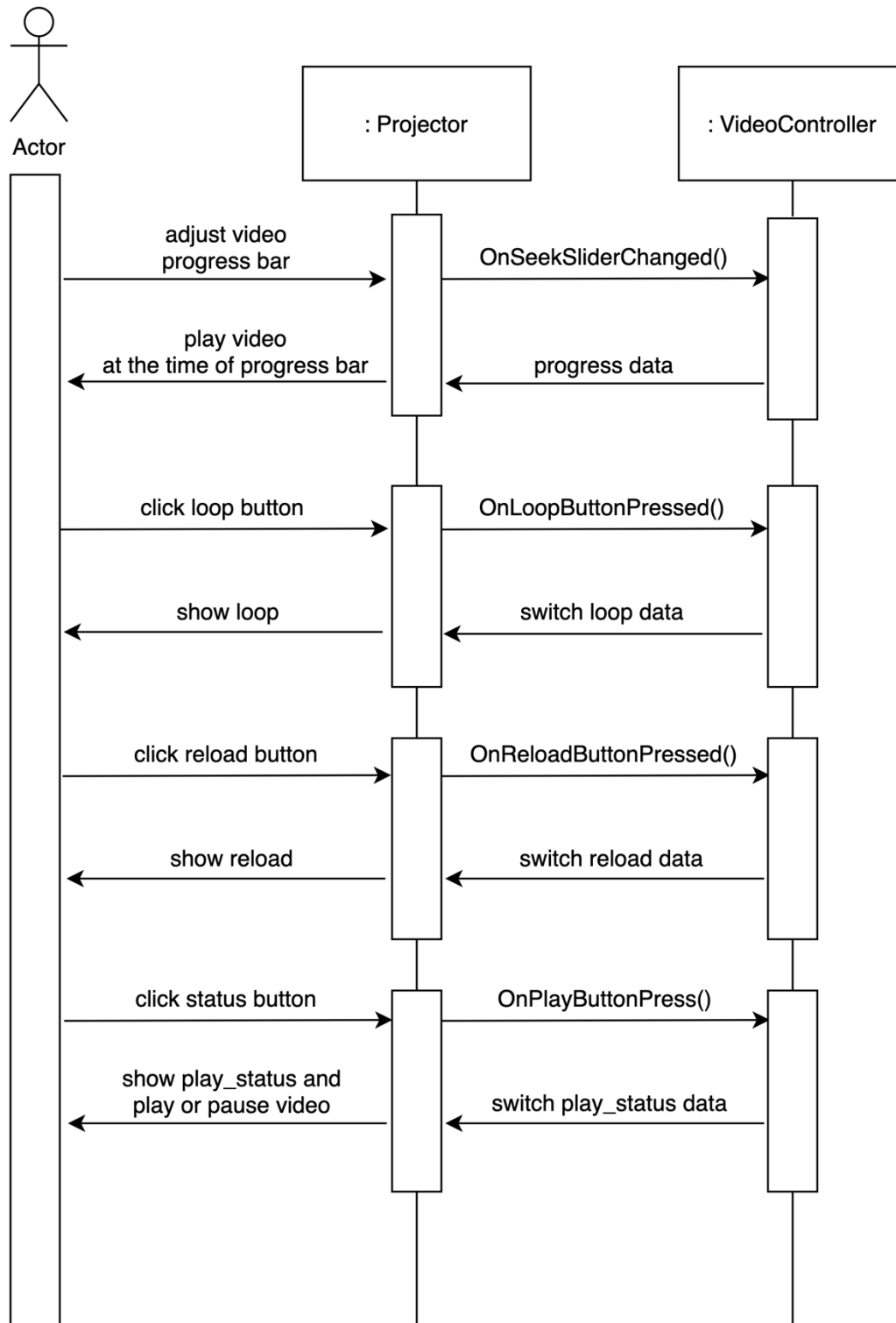
4.2.1.4. Sequence Diagram



[Figure 6] Sequence diagram – Information



[Figure 7] Sequence diagram – Volume



[Figure 8] Sequence diagram – VideoController

4.2.2. Whisper Zone

Whisper Zone 기능은 회의 중에 소수의 인원만이 따로 회의를 해야 할 경우에 지정된 장소에서 이야기를 나누면 지정된 장소(Voice Block Area)에 있는 사람들의 대화는 그 장

소 외에 있는 사람들에게는 들리지 않게 해주는 기능입니다.

4.2.2.1. Attributes

Whisper zone과 관련된 속성들이 존재합니다.

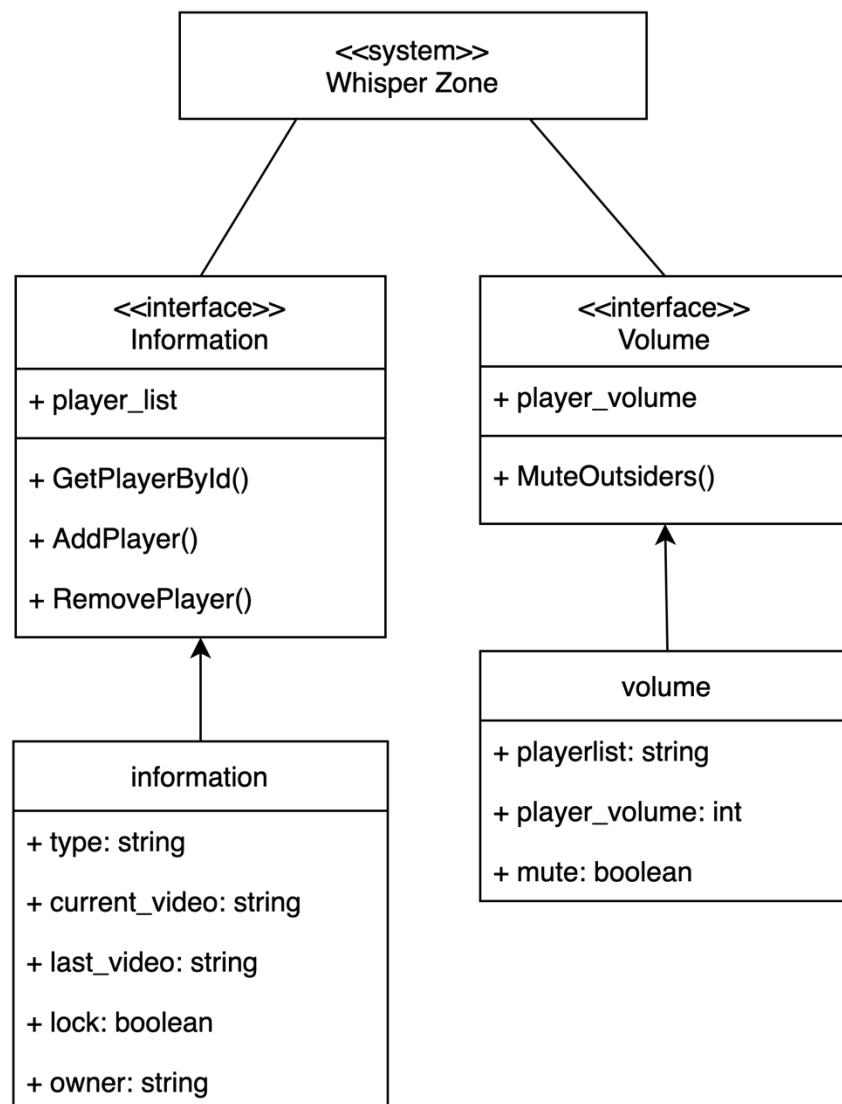
- **Voice Block Area:** 목소리가 외부와 차단되는 특정 장소입니다.
- **Player_list:** 특정 장소에 있는 사용자들을 나타냅니다.
- **Player_Volume:** 사용자들의 볼륨을 나타냅니다.

4.2.2.2. Methods

Whisper zone과 관련된 method들은 다음과 같습니다.

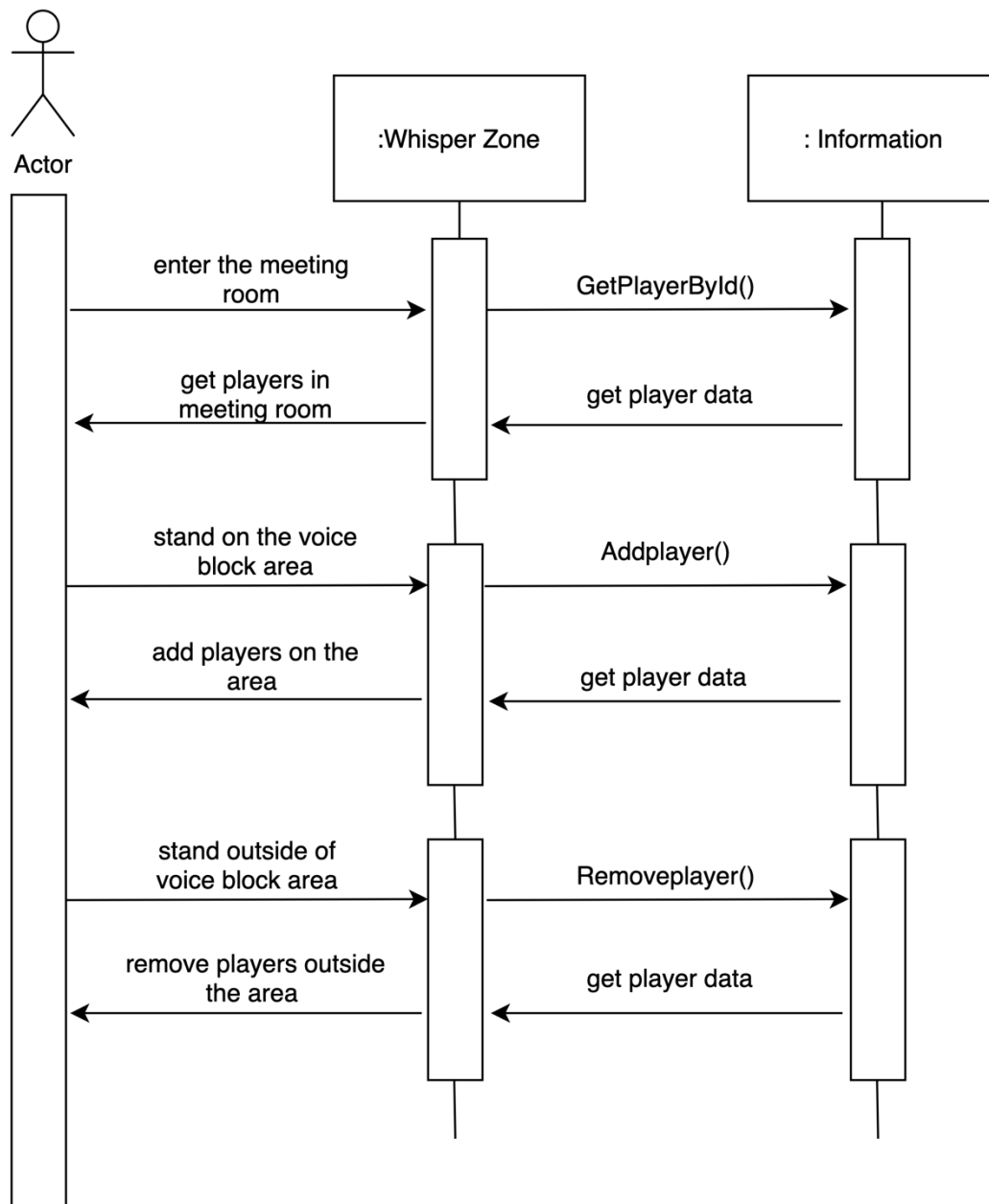
- **GetPlayerById():** 사용자 데이터를 가져와 리스트로 저장합니다.
- **AddPlayer():** 사용자 데이터를 추가합니다.
- **RemovePlayer():** 사용자 데이터를 제거합니다.
- **MuteOutsiders():** 사용자 리스트 안에 없는 사용자들을 음소거합니다.

4.2.2.3. Class Diagram

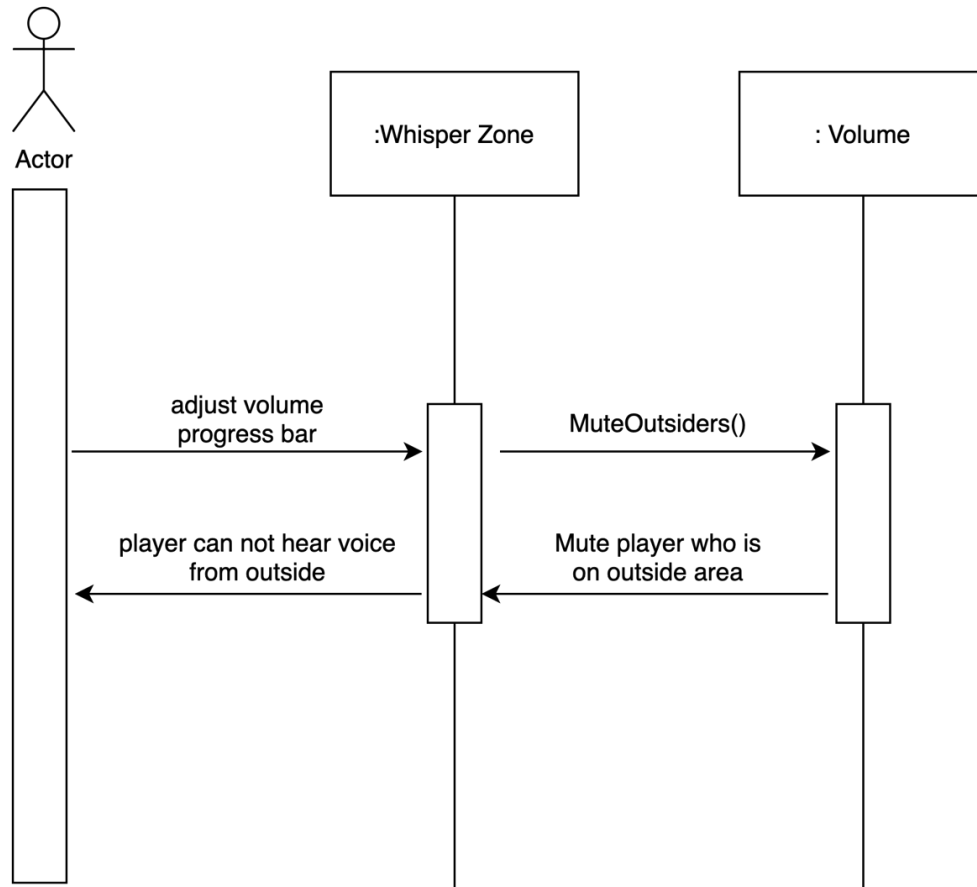


[Figure 9] Class diagram – Whisper zone

4.2.2.4. Sequence Diagram



[Figure 10] Sequence diagram – Information



[Figure 11] Sequence diagram – Volume

4.2.3. Calendar

Calendar는 년-월과 그에 속한 날짜들을 표시하는 달력입니다. 월별 이동 버튼을 조작하여 년-월을 바꾸고 이에 따라 그에 속한 날짜들이 바뀝니다. 오늘을 나타내는 날짜는 빨간색으로 표시됩니다.

4.2.3.1. Attributes

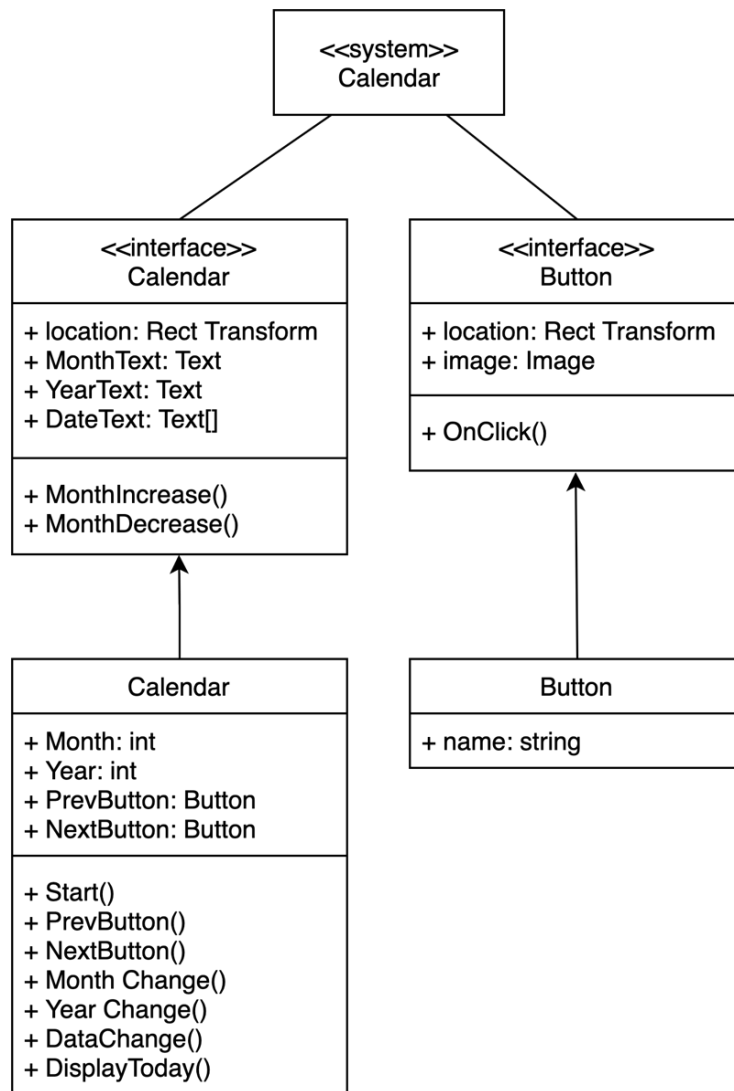
- **Month, Year:** 내부적으로 년-월을 저장합니다.
- **MonthText, YearText, DateText:** Calendar의 년-월-일을 나타내는 Text입니다.
- **DateButtons:** 달력의 각 날짜 버튼입니다.
- **StartDay:** 월의 시작 위치입니다.
- **IsNextMonthBool:** PrevButton, NextButton을 확인합니다.

4.2.3.2. Methods

- **Start():** Initial 함수로 달력을 초기화합니다.
- **MonthIncrease():** 내부 변수 월과 년을 증가시킵니다.

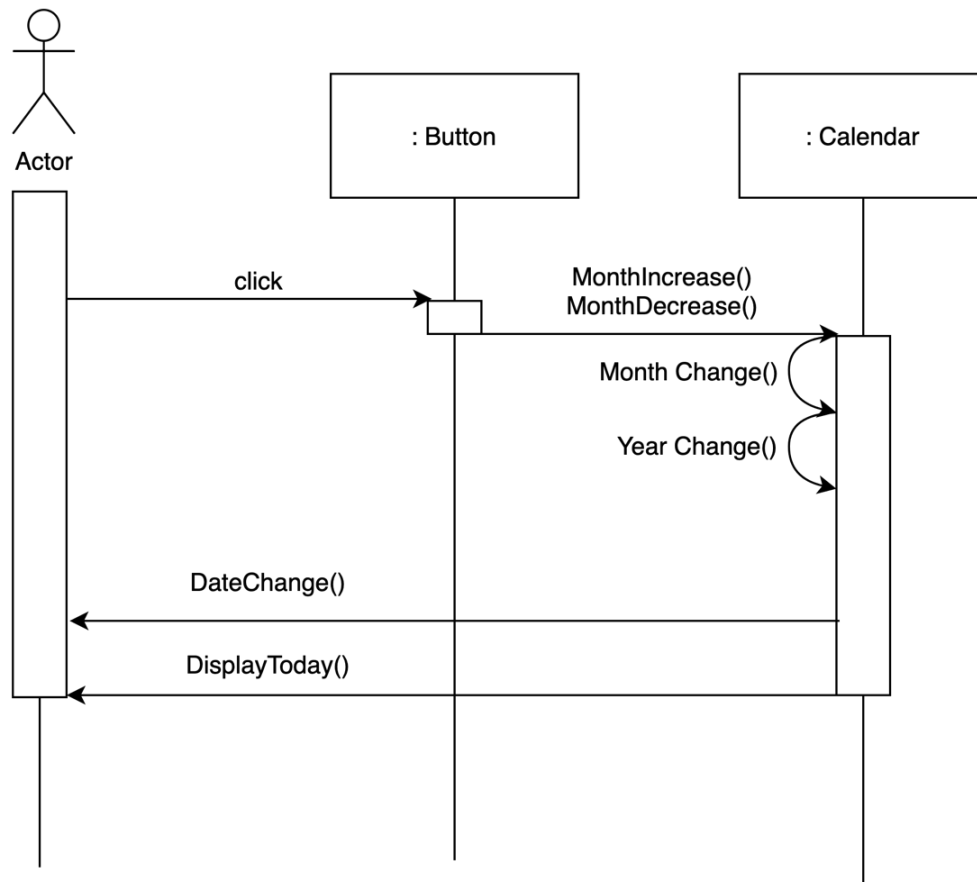
- **MonthDecrease()**: 내부 변수 월과 년을 감소시킵니다.
- **PrevButton()**: PrevButton을 눌렀을 때 실행되어, 각 User에 Event를 전달합니다.
- **NextButton()**: NextButton을 눌렀을 때 실행되어, 각 User에 Event를 전달합니다.
- **Month Change()**: Month Text를 변경하고, startDay를 설정합니다.
- **Year Change()**: Year Text를 변경합니다.
- **DataChange()**: 표시되는 날짜를 변경합니다.
- **DisplayToday()**: 현재 날짜를 표시합니다.

4.2.3.3. Class Diagram



[Figure 12] Class diagram – Calendar

4.2.3.4. Sequence Diagram



[Figure 13] Sequence diagram – Button

4.2.4. Pen

보다 원활한 의사소통을 위해 사용자들은 다양한 색상의 펜을 사용해 아무 벽에 판서를 할 수 있습니다. 펜을 들고 벽 근처에 다가가서 벽을 향해 마우스 왼쪽 버튼을 클릭하면 펜이 활성화되고 쓰기가 시작됩니다. 그리고 도중에 클릭을 해제하거나 벽에서 멀리 떨어지면 쓰기가 중지됩니다. 또한 펜을 든 채로 마우스 왼쪽 버튼을 더블 클릭하면 지우개 모드로 전환됩니다. 펜 축 부분에 동그란 모양의 지우개가 보이면 지우개 모드로 전환된 것입니다. 지우개 모드에서 마우스 왼쪽 버튼을 클릭하고 있는 동안 지우개가 활성화되고, 이 상태에서 펜으로 그린 선에 지우개를 가져다 대면 지워집니다. 다시 마우스 왼쪽 버튼을 더블 클릭해서 펜 모드로 돌아갈 수 있습니다. 각 유저가 판서하고 지우는 내용은 다른 모든 유저들에게 실시간으로 동기화됩니다.

4.2.4.1. Attributes

Pen이 가지는 속성들은 다음과 같습니다.

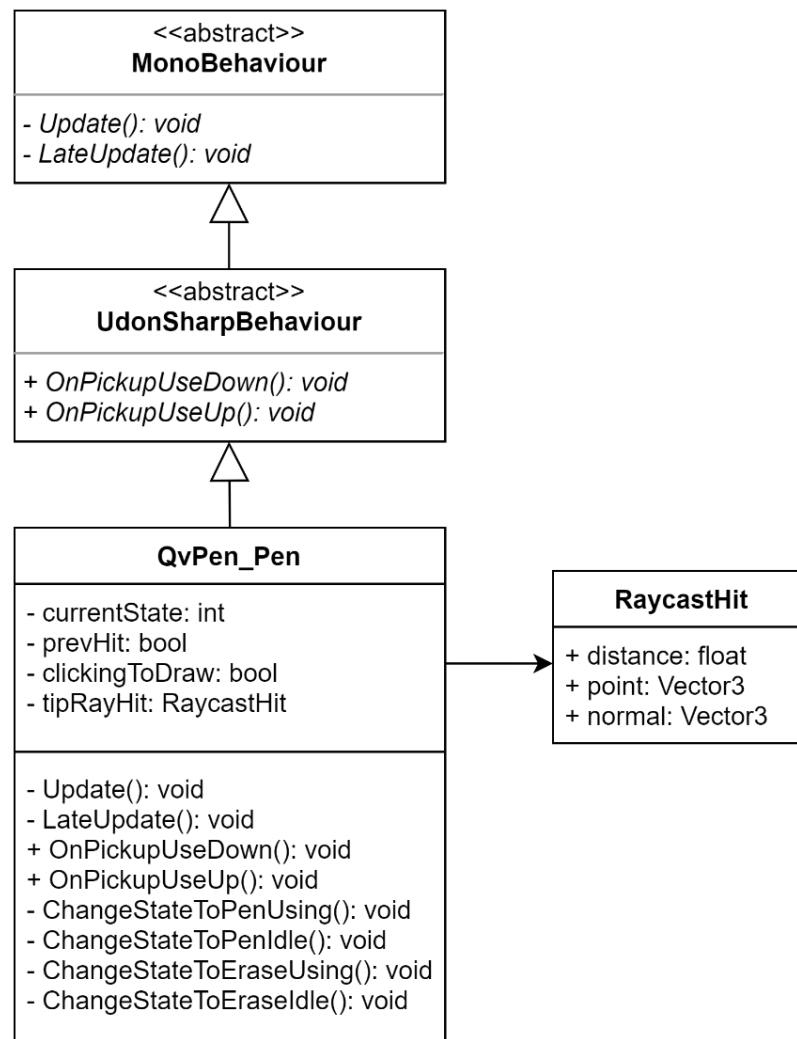
- **currentState**: Pen 오브젝트의 현재 상태를 나타냅니다. 가질 수 있는 값은 아래 네 가지가 있습니다.
StatePenIdle: 펜 모드, 펜 비활성화됨
StatePenUsing: 펜 모드, 펜 활성화됨
StateEraserIdle: 지우개 모드, 지우개 비활성화됨
StateEraserUsing: 지우개 모드, 지우개 활성화됨
- **prevHit**: 이전 프레임에서 펜이 활성화될 수 있을 만큼 벽과 충분히 가까이 있었는지를 나타냅니다.
- **clickingToDraw**: 현재 사용자가 펜을 활성화하기 위해 마우스를 누르고 있는지를 나타냅니다.
- **tipRayHit**: 펜 축에서 전방으로 뻗어 나가는 가상의 광선이 부딪히는 지점에 대한 정보를 저장합니다.

4.2.4.2. Methods

Pen이 가지는 method들은 다음과 같습니다.

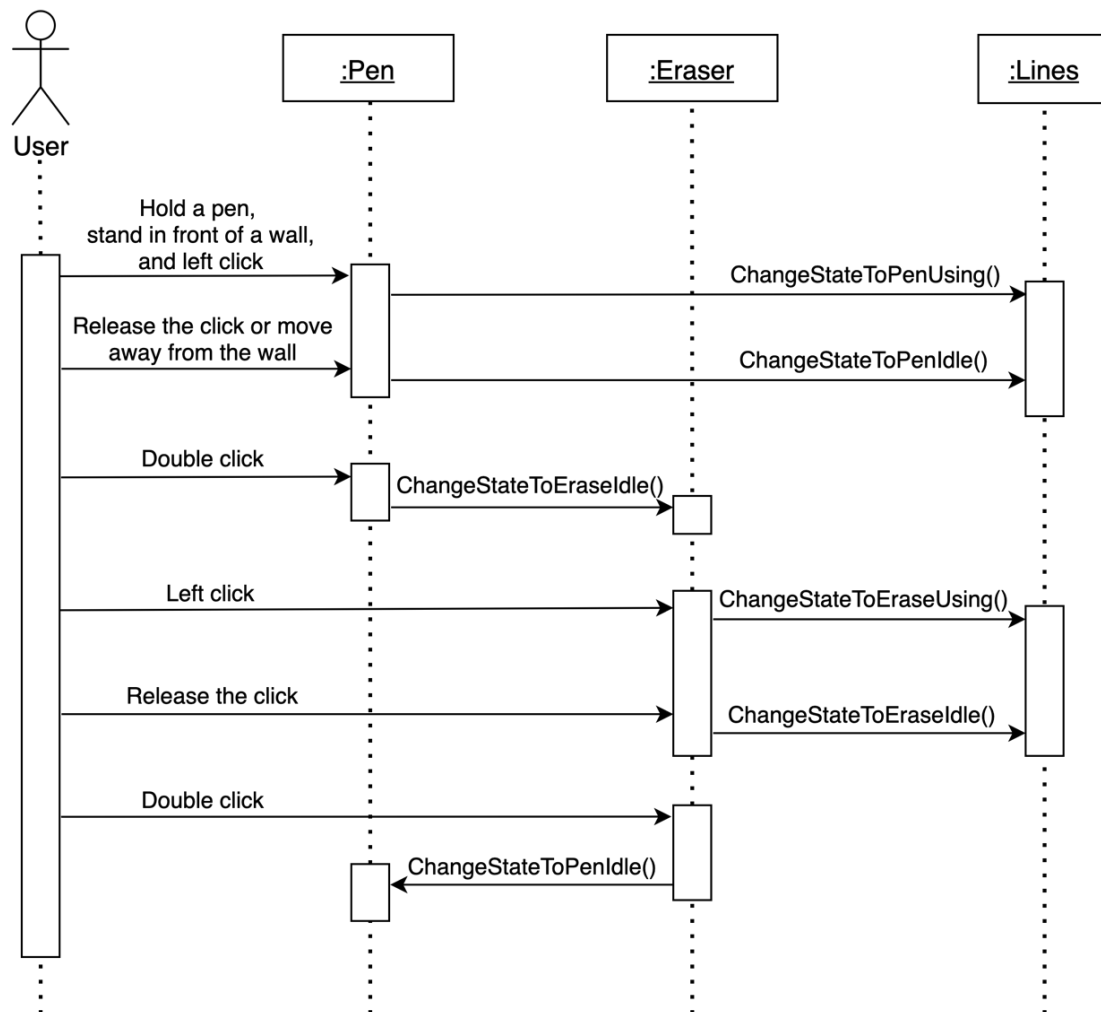
- **Update()**: 매 프레임마다 자동으로 호출되는 method입니다. 펜 축에서 전방으로 가상의 광선을 투사해서 일정 거리 내에 부딪히는 벽이 있는지 확인합니다.
- **LateUpdate()**: 매 프레임마다 Update()가 끝나고 나서 자동으로 호출되는 method입니다. Update()에서 확인한 내용을 바탕으로 그림이 그려져야 하는 위치를 계산합니다.
- **OnPickupUseDown()**: 사용자가 펜을 든 상태로 마우스 왼쪽 버튼을 클릭했을 때 자동으로 호출됩니다.
- **OnPickupUseUp()**: 사용자가 펜을 든 상태로 마우스 왼쪽 버튼을 클릭 해제했을 때 자동으로 호출됩니다.
- **ChangeStateToPenUsing()**: 활성화된 펜 모드로 전환합니다.
- **ChangeStateToPenIdle()**: 비활성화된 펜 모드로 전환합니다.
- **ChangeStateToEraseUsing()**: 활성화된 지우개 모드로 전환합니다.
- **ChangeStateToEraserIdle()**: 비활성화된 지우개 모드로 전환합니다.

4.2.4.4. Class Diagram



[Figure 14] Class diagram – Pen

4.2.4.5. Sequence Diagram



[Figure 15] Sequence diagram – Pen

5. Testing Plan

5.1. Objectives

5단원 Testing Plan에서는 요구사항 명세서에서 기술한 내용을 바탕으로 전체 시스템이 잘 실행되는지 확인합니다. 이러한 Testing Plan을 통해 저희는 프로그램에서 발생할 수 있는 잠재적 오류 및 결함을 감지하고 프로그램의 완벽한 작동과 사용자 경험을 미리 확인할 수 있습니다.

5.2. Testing Policy

Testing Plan은 development testing, release testing, user testing 세 단계로 구성됩니다. 다음은 각각의 testing의 정의와 과제들을 서술하고 있습니다.

5.2.1. Development Testing

Development testing은 시스템을 개발하는 과정에서 진행하는 테스트 단계입니다. 개발 단계가 나아갈 때마다, 만들어진 모든 구성 요소를 테스트하는 것으로 구성 요소가 제대로 작동하는지를 확인하고자 합니다. 이 단계에서는 component testing, system testing 그리고 acceptance testing을 순차적으로 진행합니다.

Component testing은 개별 구성 요소들을 독립적으로 테스트하며, component란 기능, 객체 혹은 entity 모음입니다. System testing은 Component들이 모여 하나의 전체 system을 테스트합니다. 기능적인 요소뿐만 아니라 비기능적인 요구사항을 만족하는지 확인합니다. Acceptance testing에서는 구현한 기능이나 객체들이 제대로 동작하는지 확인합니다.

5.2.2. Release Testing

최종적으로 완성된 시스템을 deploy하기 전에 실시하는 테스트입니다. 시스템이 사용자의 요구사항을 충족하는지, 서비스를 개발 의도에 맞게 제공하는지, 정상적으로 기능이 작동하는지 등을 확인합니다. 본 과정에서는 VRCSDK의 builder기능을 활용하여 테스트 월드를 빌드하여 위의 내용들을 확인할 수 있습니다.

5.2.3. User Testing

사용자의 실제 사용 환경에서 시스템을 테스트합니다. 실제 사용자들을 통해 개발한 VRChat world가 사용하기 쉬운지, 서비스의 활용성이나 성능은 뛰어난지 등을 평가합니다.

5.3. Test Case

5.3.1. Pen

- a) User: 미팅룸에 있는 펜을 클릭합니다.
- b) 시스템 동작: 사용자의 아바타가 펜을 집어 올릴 수 있습니다. 손에 든 펜으로. 미팅룸 내에 Screen 위에 그림을 그릴 수 있습니다.

5.3.2. Projector

- a) User: 비디오 URL을 입력합니다.
- b) 시스템 동작: URL에 있는 유튜브 등의 비디오를 미팅룸 내에 Screen 위에 재생시킬 수 있다.

5.3.3. Calendar

- a) User: 달력에 있는 화살표를 클릭합니다.
- b) 시스템 동작: 화살표의 방향에 따라 월을 바꾸고 그 달의 날짜를 확인 할 수 있습니다.

5.3.4. Whisper Zone

- a) User: 미팅룸 바닥의 색이 다른 특정 장소로 이동합니다.
- b) 시스템 동작: 특정 장소 안과 밖의 소리는 차단되어 서로 들리지 않습니다.

6. Development Plan

6.1. Objectives

이 장에서는 프로그램 개발을 위해 사용한 개발 도구와 환경을 설명합니다.

6.2. Frontend Environment

6.2.1. Autodesk Inventor



[Figure 16] Autodesk Inventor logo

Autodesk Inventor는 컴퓨터 지원 설계(CAD) 소프트웨어로, 3D 기반의 설계, 시각화, 시뮬레이션에 사용됩니다. Vroom에 배치하기 위한 개개의 오브젝트들의 모델링을 위해서 사용됩니다.

6.2.2. Unity



[Figure 17] Unity logo

Unity는 게임 개발 환경을 제공하는 게임 엔진인 동시에 가상 현실 콘텐츠 제작을 위한 소프트웨어입니다. Vroom의 방 전체 구조와 오브젝트의 배치를 위해 사용됩니다.

6.2.3. VRChat



[Figure 18] VRChat logo

VRChat은 대규모 오픈 월드 멀티 플랫폼 게임으로, VR 기기 혹은 데스크탑을 사용하여 3D로 모델링된 가상 현실을 즐길 수 있는 프로그램입니다. 유니티로 개발한 모델을 실행, 테스트하기 위해 사용됩니다. 또한 VRoom 월드 내의 이용자들에게 적용되는 특수 기능들과 내부 오브젝트의 상호작용 기능들을 추가하기 위해 VRChat이 개발한 프로그래밍 언어인 Udon을 사용하였습니다.

6.3. Backend Environment

6.3.1. GitHub



[Figure 19] GitHub logo

GitHub는 분산 버전 관리 툴인 Git을 지원하는 웹 서비스입니다. 여러 팀원이 함께 하나의 프로젝트를 공유하며 개발하기 위해 필수적인 코드 통합, 버전 관리 등의 기능을 위하여 사용됩니다.

6.3.2. VRchat



[Figure 20] VRChat logo

VRRoom을 이용하는 이용자들은 VRChat의 서버 공간 안에서 자유롭게 다른 이용자들을 만나고, 회의할 수 있습니다.

6.4. Open source

저희는 비디오 플레이어를 구현하기 위해 오픈소스 2가지를 사용했습니다.

6.4.1. QvPen¹

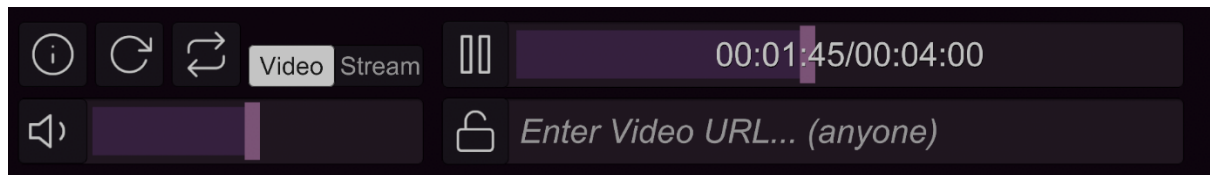
QvPen은 VRChat world에서 3차원 공간에 그림을 그리는 데 사용할 수 있는 펜과 지우개를 UdonSharp으로 구현한 오픈소스입니다. VRRoom world에서 사용하는 Pen은 QvPen의 소스코드를 수정해서 3차원 공간이 아닌 벽면에만 그림을 그릴 수 있도록 구현하였습니다. QvPen은 라이선스가 없는 오픈소스입니다.

6.4.2. BetterAudio²

BetterAudio는 사용자 오디오와 관련된 기능을 제공하는 UDON 기반 VRChat 오픈소스입니다. 이 오픈소스에는 플레이어의 음성, 아바타 사운드, 리버브, 음성/아바타 오디오 페섹 및 방향성을 자동으로 제어하는 도구가 포함되어 있습니다. Whisper zone기능을 구현하는데 사용되었습니다. MIT License 오픈소스입니다.

6.4.3. UdonSharpVideo³

비디오 플레이어와 관련된 기능을 제공하는 오픈소스입니다. Udon과 UdonSharp을 이용하여 제작되었습니다. 일반 비디오와 라이브 스트리밍 기능을 제공합니다. 구성 요소로는 권한 설정 버튼, 비디오 재생 bar, 정지/재생 버튼, 다시 재생하기 버튼, 비디오/스트리밍 선택 버튼, 음량 조절 bar, 잠금 버튼, URL 입력 등이 있습니다. 다양한 색깔의 테마를 제공하고, 개인 커스텀 또한 가능합니다. MIT License 오픈소스입니다.



[Figure 21] UdonSharpVideo image

¹ <https://github.com/ureishi/QvPen>

² <https://github.com/Guribo/BetterAudio>

³ <https://github.com/MerlinVR/USharpVideo>

6.4.4. UdonSharp⁴

UdonSharp는 C#을 Udon 어셈블리어로 컴파일 하는 컴파일러로, C#언어로 작성된 코드를 Udon-Graph로 바꿔주는 기능을 하는 오픈소스입니다. UdonSharpVideo를 사용하기 위해서 필요한 package입니다. MIT License 오픈소스입니다.

6.5. Constraints

본 프로그램은 이 문서에 언급된 내용을 기반으로 설계 및 구현됩니다. 기타 세부 사항은 개발자가 선호하는 방향을 선택하여 설계 및 구현하되, 다음 사항을 준수하였습니다.

- 이미 검증된 기술만을 사용한다.
- 별도의 라이선스 혹은 로열티가 필요한 소프트웨어를 사용하지 않는다. (단, 해당 소프트웨어 이외의 대안이 없을 경우 허용한다.)
- 가급적 오픈 소스 소프트웨어를 사용한다.
- 전체 시스템의 성능을 향상시킬 수 있는 방향을 선택한다.
- 시스템 비용과 유지 보수에 소요되는 비용을 고려하여 선택한다.
- 시스템의 미래 확장성을 고려하여 선택한다.
- 보다 사용자 친화적인 방향을 선택한다.
- 향후 유지 보수를 고려하여 충분한 주석을 작성한다.
- 개발 시 Windows 10, Unity 2019.4.30f1 이상의 버전 환경에서 개발한다.

6.6. Assumptions and Dependencies

본 문서의 모든 시스템은 Windows 10 및 VRChat을 기반으로 설계 및 구현되었다는 가정 하에 작성되었습니다. 따라서 모든 콘텐츠는 다른 운영체제 혹은 버전에서는 적용되지 않을 수 있습니다.

⁴ <https://github.com/MerlinVR/UdonSharp>

7. Supporting Information

7.1. Software Design Specification

0| software design specification는 IEEE Recommendation 에 따라 작성되었습니다 (IEEE Recommended Practice for Software Design Description, IEEE-Std-1016).

7.2. Document History

[Table 1] Document History

Date	Version	Description	Writer
2021. 11. 18.	0.1	Initial Draft	All Members
2021. 11. 19	0.2	Add: 4.2.2. Whisper Zone	Hong Seongjun
2021. 11. 19	0.3	Add: 3. System Architecture: Overall	Seo Chaeyeon
2021. 11. 19	0.4	Add: 4.2.1.5. Open Source	Seo Chaeyeon
2021. 11. 20	0.5	Add: 4.2.3. Calendar	Kim Jinwoong
2021. 11. 20	0.6	Revision: 1. Preface, 4.2.1. Projector	Kim Minhee
2021. 11. 20	0.7	Add: 4.2.4. Pen	Kim Hongbeen
2021. 11. 20	0.8	Revision: Pictures of 3. Sys. Arch.	Seo Chaeyeon
2021. 11. 20	0.9	Revision: 4.2.2 Whisper zone	Hong Seongjun
2021. 11. 21	0.10	Revision: 4.2.4.2. Methods	Kim Hongbeen
2021. 11. 21	1.0	Basic Draft (Chapter Rebased)	Kim Jinwoong
2021. 11. 21	1.1	Revision: 6.2.3., 6.3.2. VRChat Add: 6.4. Open Source	Lee Jiwoo
2021. 11. 21	1.2	Add: 2.5 References	Lee Hyewon
2021. 11. 21	1.3	Final Draft	Kim Jinwoong