



Platform for School Event

Software Design Specification

2021.11.21

Team3

강보영

김현민

신세린

이산하

채세이

최규리

Contents

1. Preface

1.1 Readership

1.2 Document Structure

1.2.A Preface

1.2.B Introduction

1.2.C Overall System Architecture

1.2.D System Architecture : 관리자 인증

1.2.E System Architecture : 실시간 스트리밍 스크린

1.2.F System Architecture : 축제 분위기 전환

1.2.G System Architecture : 게시판

1.2.H System Architecture : 이벤트 부스

1.2.I System Architecture : 축제용 아바타

1.2.J 비디오-유튜브 연결 Design

1.2.K Testing Plan

1.2.L Development Plan

1.2.M Index

1.3 References

2. Introduction

2.1 Objectives

2.2 Applied Diagrams

2.2.A Unified Modeling Language

2.2.B Class Diagram

2.2.C Sequence Diagram

2.2.D State Diagram

2.2.E Deployment Diagram

2.2.F Use Case Diagram

2.2.G ER Diagram

2.3 Applied Tool

2.3.A Draw.io

2.3.B ERDPlus

2.4 Project Scope

3. Overall System Architecture

3.1 Objectives

3.2 System Organization

3.3 관리자 인증

3.4 실시간 스트리밍 스크린

3.5 축제 분위기 전환

3.6 게시판

3.7 이벤트 부스

3.8 축제용 아바타

4. System Architecture : 관리자 인증

4.1 Objectives

4.2 Overall Architecture

4.3 Detailed Architecture

4.3.A Class Diagram

4.3.B Sequence Diagram

5. System Architecture : 실시간 스트리밍 스크린

5.1 Objectives

5.2 Overall Architecture

5.3 Detailed Architecture

5.3.A Class Diagram

5.3.B Sequence Diagram

5.3.C State Diagram

6. System Architecture : 축제 분위기 전환

6.1 Objectives

6.2 Overall Architecture

6.3 Detailed Architecture

6.3.A Class Diagram

6.3.B Sequence Diagram

6.3.C State Diagram

7. System Architecture : 게시판

7.1 Objectives

7.2 Overall Architecture

7.3 Detailed Architecture

7.3.A Class Diagram

7.3.B Sequence Diagram

8. System Architecture : 이벤트 부스

8.1 Objectives

8.2 Overall Architecture

8.3 Detailed Architecture : 블라인드 소개팅 부스

8.3.A Class Diagram

8.3.B Sequence Diagram

8.4 Detailed Architecture : 친목도모 부스

8.4.A Class Diagram

8.4.B Sequence Diagram

8.5 Detailed Architecture : 미니게임 부스

8.5.A Class Diagram

8.5.B Sequence Diagram

9. System Architecture : 축제용 아바타

9.1 Objectives

9.2 Overall Architecture

9.3 Detailed Architecture

9.3.A Class Diagram

9.3.B Sequence Diagram

10. 비디오-유튜브 연결 Design

10.1 Objectives

10.2 ER Diagram

10.2.A Entities

10.2.B Relation

10.3 Relational Schema

11. Testing Plan

11.1 Objectives

11.2 Testing Policy

11.2.A Development Testing

11.2.A.1 Usability

11.2.A.2 Security

11.2.A.3 Reliability

11.2.B Release Testing

11.2.C User Testing

11.2.D Testing Case

12. Development Plan

12.1 전체 시스템 Integration

12.2 실시간 스트리밍 스크린

12.3 축제 분위기 전환

12.4 게시판

12.5 이벤트 부스

12.6 축제용 아바타

13. Index

13.1 Tables

13.2 Figures

13.3 Diagrams

1 Preface

1.1 Readership

본 소프트웨어 디자인 명세서를 읽을 독자는 해당 시스템을 개발 및 유지보수하는 개발자들(2021 년도 제 2 학기 소프트웨어공학개론 3 조)와 소프트웨어공학개론의 교수님, 조교님들이다.

1.2 Document Structure

1.2.A Preface

Preface 에서는 본 문서의 독자 정보와 그 범위, 문서의 목적과 구조에 대한 내용에 대해 기술한다.

1.2.B Introduction

Introduction 에서는 시스템 설계에 사용된 UML(Unified Modeling Language)의 종류와 그 예시에 대해 설명하고, diagram 작성에 사용된 프로그램에 대해 소개한다.

1.2.C Overall System Architecture

Overall System Architecture 에서는 본 프로젝트의 시스템 아키텍처에 대한 전반적인 개요를 보여준다. 각 세부기능 별로 block diagram 을 활용하여 한눈에 시스템을 파악할 수 있도록 하였다.

1.2.D System Architecture : 관리자 인증

System Architecture : 관리자 인증에서는 관리자 인증 시스템에 대해 앞서 설명한 시스템 아키텍처에 대한 보다 자세한 설명을 제공한다. Class diagram, sequence diagram 을 사용하여 시스템의 구조와 상호작용에 대해 기술한다.

1.2.E System Architecture : 실시간 스트리밍 스크린

System Architecture : 실시간 스트리밍 스크린에서는 실시간 스트리밍 스크린 시스템에 대해 앞서 설명한 시스템 아키텍처에 대한 보다 자세한 설명을 제공한다. Class diagram, sequence diagram, state diagram 을 사용하여 시스템의 구조와 상호작용에 대해 기술한다.

1.2.F System Architecture : 축제 분위기 전환

System Architecture : 축제 분위기 전환에서는 축제 분위기 전환 시스템에 대해 앞서 설명한 시스템 아키텍처에 대한 보다 자세한 설명을 제공한다. Class diagram, sequence diagram, state diagram 을 사용하여 시스템의 구조와 상호작용에 대해 기술한다.

1.2.G System Architecture : 게시판

System Architecture : 게시판에서는 게시판 시스템에 대해 앞서 설명한 시스템 아키텍처에 대한 보다 자세한 설명을 제공한다. Class diagram, sequence diagram 을 사용하여 시스템의 구조와 상호작용에 대해 기술한다.

1.2.H System Architecture : 이벤트 부스

System Architecture : 이벤트 부스에서는 이벤트 부스 시스템에 대해 앞서 설명한 시스템 아키텍처에 대한 보다 자세한 설명을 제공한다. Class diagram, sequence diagram 을 사용하여 시스템의 구조와 상호작용에 대해 기술한다.

1.2.I System Architecture : 축제용 아바타

System Architecture : 축제용 아바타에서는 축제용 아바타 시스템에 대해 앞서 설명한 시스템 아키텍처에 대한 보다 자세한 설명을 제공한다. Class diagram, sequence diagram 을 사용하여 시스템의 구조와 상호작용에 대해 기술한다.

1.2.J 비디오-유튜브 연결 Design

비디오-유튜브 연결 Design 에서는 본 프로젝트의 실시간 스트리밍 비디오 기능에 대해 기술한다. ER diagram 을 사용하여 각 entity 가 어떤 식으로 상호작용하며 해당 기능을 하는지 보여준다. 또한, relational schema 를 통해 해당 시스템을 간결하게 보여준다.

1.2.K Testing Plan

Testing Plan 에서는 개발된 시스템이 요구사항 명세서에 제시된 다양한 요구사항에 부응하는지, 오류는 없는지 확인하기 위한 testing 계획에 대해 기술한다.

1.2.L Development Plan

Development Plan 에서는 향후 시스템 개발이 어떻게 진행될지에 대한 계획과 구현에 사용되는 다양한 라이브러리 및 SDK 에 대해 소개한다.

1.2.M Index

본 문서의 목차와 본문에 포함된 표와 그림에 대한 인덱스를 정리한다.

1.3 References

[1] Team 3. "Software Requirement Specification". SKKU, Last Modified: Sep. 26, 2021.

Available :

https://github.com/skkuse/2021fall_41class_team3/blob/main/docs/Team3_%EC%9A%A

94%EA%B5%AC%EC%82%AC%ED%95%AD_%EB%AA%85%EC%84%B8%EC%84%9C.p
df

[2] The VRChat Documentation Hub.

Available : <https://docs.vrchat.com/>

[4] "UML: 클래스 다이어그램과 소스코드 매핑". [Online].

Available : <https://www.nextree.co.kr/p6753/>

[5] "[UML] 시퀀스 다이어그램 이해하기". [Online].

Available : <https://sabarada.tistory.com/84?category=800100>

[6] "(UML) 유스케이스 다이어그램 – Usecase Diagram". [Online].

Available : <https://googry.tistory.com/2>

[7] Draw.io. [Online].

Available : <https://app.diagrams.net/>

[8] ERDPlus. [Online].

Available : <https://erdplus.com/>

2 Introduction

2.1 Objectives

이 챕터에서는 해당 디자인 명세서 작성에 사용된 다이어그램과 톨에 대한 개괄적인 설명을 할 예정이다.

2.2 Applied Diagrams

2.2.A Unified Modeling Language

Unified Modeling Language (UML)은 도메인, 즉 시스템으로 해결하고자 하는 목표를 모델로 표현해주는 대표적인 모델링 언어이다. 이러한 UML 을 사용하여 소프트웨어를 설계할 수 있으며, 이는 필요에 의해 사용된다. 일반적으로 UML 은 의사소통과 설계 논의를 위해 사용되거나, 전체 시스템의 구조 및 클 sub-system들의 의존성을 파악할 때 사용되거나, 유지보수를 위한 설계 back-end 문서 제작을 위해 사용된다.

각 UML 에는 개발자와 사용자의 의사소통을 위해 다양한 정의와 표현들이 사용될 수 있다. 다양한 종류의 UML 이 존재하는데, Class Diagram, Sequence Diagram, Package Diagram, Deployment Diagram, State Diagram, ER Diagram 이 그 예시이다.

2.2.B Class Diagram

Class Diagram 은 정적 다이어그램으로 클래스의 구성요소 및 sub-system 간의 관계를 표현하는 대표적인 UML 이다. Class Diagram 을 통해 sub-system 들 간의 의존관계를 명확히 보여줄 수 있으며, 순환 의존이 발생하는 지점을 찾아내서 어떻게 이 순환고리를 깰 수 있을지 결정할 수 있게 해준다.

Class Diagram 은 2 가지 용도로 사용될 수 있다. 첫번째는 문제 해결을 위한 도메인 구조를 나타내어 보이지 않는 도메인 안의 개념과 같은 추상적인 개념을 기술하기 위해

사용될 수 있다. 두번째는 소프트웨어의 설계 혹은 완성된 소프트웨어의 구현 설명을 목적으로 사용할 수 있다. 우리는 이번 디자인 명세서에서 각 sub-system 들의 설계와 sub-system 내의 클래스들 간의 상호작용 및 의존성을 나타내기 위해 class diagram 을 사용했다.

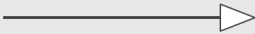



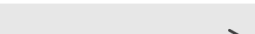




관계	UML 표기
Generalization (일반화)	
Realization (실체화)	
Dependency (의존)	
Association (연관)	
Directed Association (직접연관)	
Aggregation (집합, 집합연관)	
	
Composition (합성, 복합연관)	
	

Figure 1. Class Diagram 의 class 관계 종류

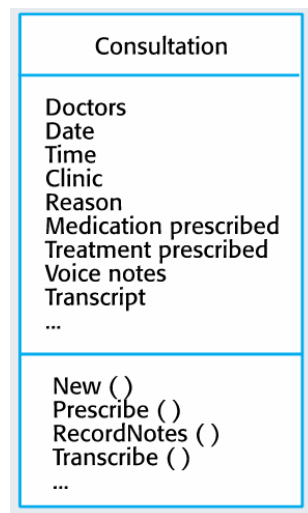


Figure 2. Class Diagram 의 class 예시

2.2.C Sequence Diagram

Sequence Diagram 은 어떠한 순서로 어떤 객체들과 어떻게 상호작용했는지를 표현하는 다이어그램이다. Sequence Diagram 은 현재 존재하는 시스템이 어떠한 시나리오로 움직이고 있는지를 나타내는데 장점을 가지고 있으며 개체와 시간으로 표현된다. 특히, sequence diagram 을 사용하면 API 등의 use case 를 구체적으로 알 수 있다.

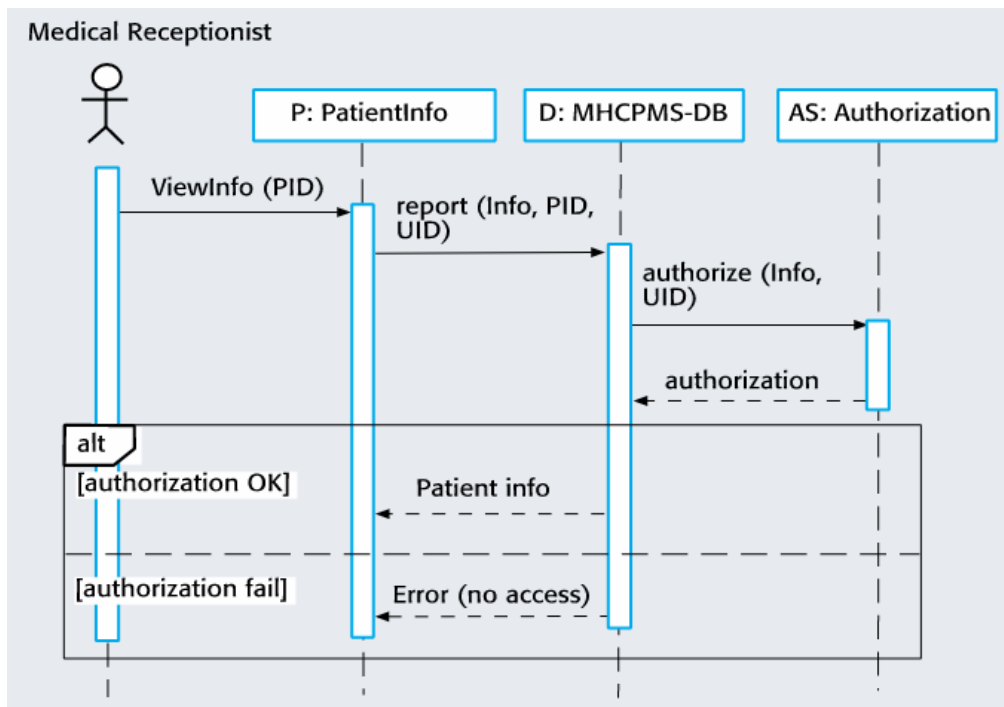


Figure 3. Sequence Diagram 예시

2.2.D State Diagram

State Diagram 에서는 시스템에 일어나는 변화를 state 로 표현한다. 사건이나 시간에 따라 시스템 내의 객체들이 자신의 상태(state)를 바꾼다. 이러한 state diagram 을 통해 시스템의 변화를 잡아낼 수 있고, 객체의 state transfer, state transfer sequence 의 시작점과 종료 시점 표시 등을 할 수 있다.

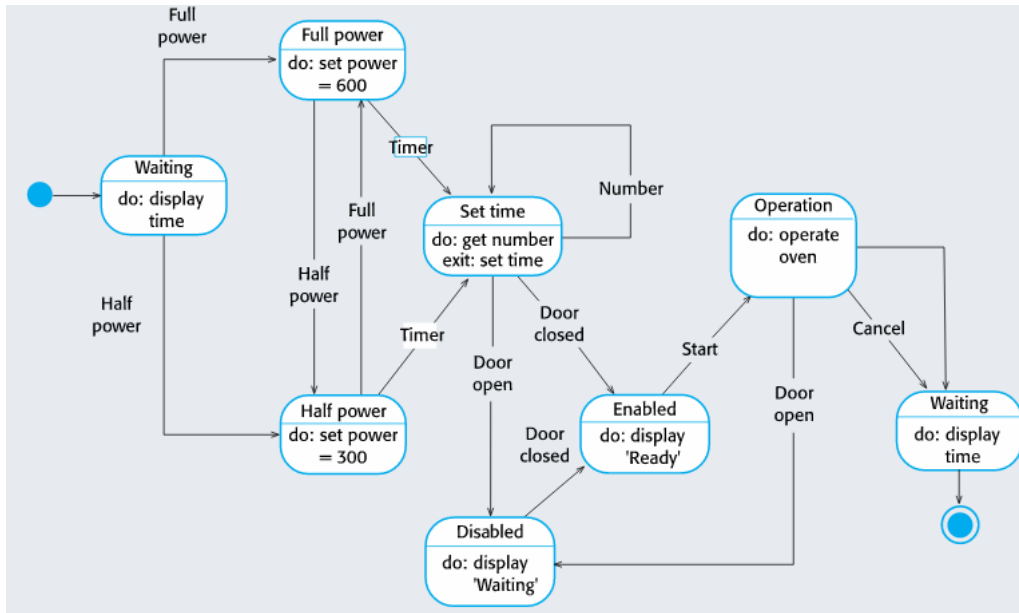


Figure 4. State Diagram 예시

2.2.E Deployment Diagram

Deployment Diagram 은 객체 지향 시스템의 물리적 측면을 모델링하는 데 사용되는 구조 다이어그램이다. Deployment Diagram 은 처리 노드의 구성과 노드에서 금지하는 구성요소를 보여준다. 임베디드, 클라이언트/서버 및 분산 시스템을 시각화, 지정 및 문서화하고 포워드 및 리버스 엔지니어링을 통해 실행 가능한 시스템을 관리하는 데 중요하다.

2.2.F Use Case Diagram

Use Case Diagram 은 시스템과 사용자의 상호작용을 다이어그램으로 표현한 것으로 사용자의 관점에서 시스템의 서비스 혹은 기능 및 그와 관련한 외부 요소를 보여준다. 사용자가 시스템 내부에 있는 기능 중에 어떤 기능을 사용할 수 있는지 나타내며 Use Case Diagram 을 사용함으로써 고객과 개발자가 요구사항에 대한 의견을 조율할 수 있다.

Use Case Diagram 은 프로젝트에 대한 요구사항을 정의하고 세부기능을 분석하며 개발 범위를 정할 때 작성한다.

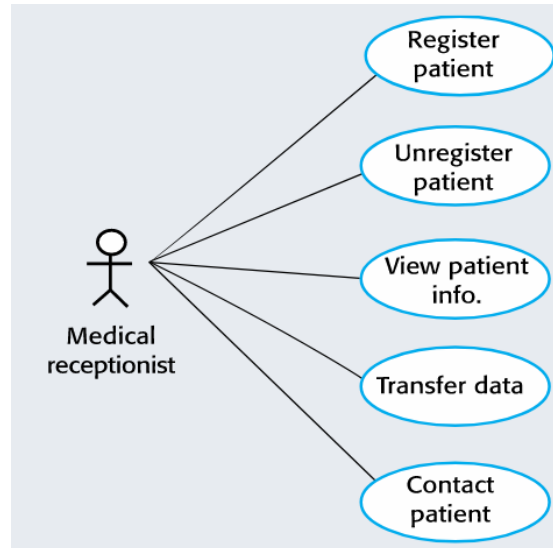


Figure 5. Use Case Diagram 예시

2.2.G ER Diagram

ER Diagram 은 Entity Relationship Diagram 으로 존재하고 있는 것(Entity)들의 관계(Relationship)을 나타낸 diagram 이다. ER 다이어그램은 데이터베이스에 있는 entity 집합의 관계를 설명한다. 일반적으로 데이터베이스의 논리적 구조 간의 관계를 표시하여 데이터베이스 디자인을 스케치하는 데 사용된다. 해당 디자인 명세서에서는 영상 스트리밍 시 유튜브 플랫폼의 비디오와 VRChat 상에서의 비디오 플레이어의 관계에 대해 설명하기 위해 ER diagram 을 사용했다.

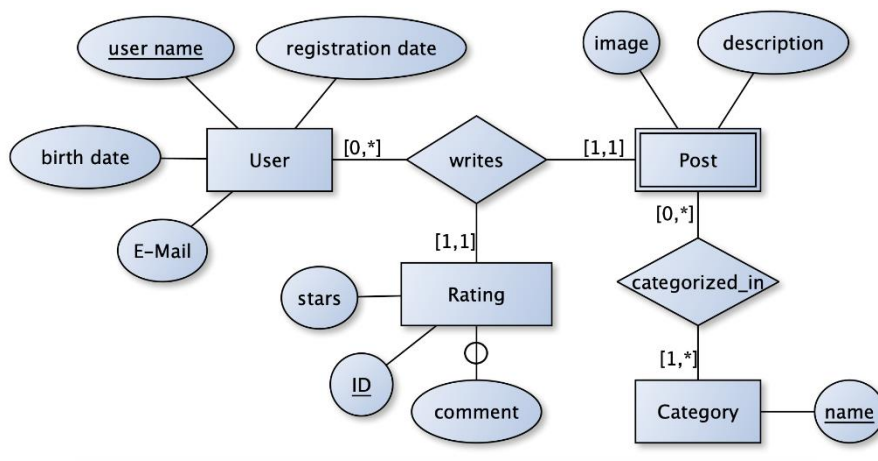


Figure 6. ER Diagram 예시

2.3 Applied Tool

2.3.A Draw.io

Draw.io 는 무료 순서도 소프트웨어이다. 각 다이어그램 종류에 따른 기존 구조가 존재하며 사용법이 직관적이다. 해당 디자인 명세서에서는 ER diagram 을 제외한 다른 다이어그램들 제작에 활용되었다.



Figure 7. Draw.io

2.3.B ERDPlus

ERDPlus 는 ER diagram 을 간단하게 제작할 수 있는 온라인 툴이다. ER diagram 을 제작하는 데에 특화되었기 때문에 draw.io 대신 ERDPlus 를 사용하여 ER diagram 을 제작했다.



Figure 8. ERDPlus

2.4 Project Scope

본 시스템은 VR Chat 기반으로 작동한다. VR Chat 은 오픈 월드 멀티 플랫폼 게임으로 자유롭게 다양한 사람들이 만들어 놓은 월드에 들어가 다른 사람과 대화하고 미니게임을 즐길 수 있는 플랫폼이다. 유니티 엔진을 통해 학교 축제 플랫폼을 만들고 이곳에서 축제를 즐길 수 있는 구조이다.

축제 공간은 크게 광장과 이벤트 부스로 이루어져 있다. 광장에는 메인 스테이지와 메인 스크린이 존재한다. 메인 스크린을 통해 실시간 스트리밍 영상을 보거나 관리자가 준비한 콘텐츠 감상이 가능하다. 이벤트 부스에서는 부스 진행자가 준비한 미니게임과 같은 다양한 콘텐츠를 즐길 수 있는 공간이다. 게시판 투표 이벤트에도 참여할 수 있다.

학생들은 아바타로 접속해 전 공간에서 다양한 사람들과 이모티콘 과 보이스를 통해 소통할 수 있다. 추가적으로 관리자는 축제 하늘과 BGM 등을 이용해 축제의 분위기를 바꿀 수 있으며, 메인 스크린과 미니 스크린에 영상을 선택하여 띄울 수도 있다. 또한, 이벤트 부스의 미니 게임을 관리하고 게시판에 투표를 올리는 등의 콘텐츠 관리를 할 수 있다.

3 Overall System Architecture

3.1 Objectives

이 챕터를 통해 본 시스템의 전체 구조에 대해 설명하고 각 서브시스템(컴포넌트) 사이의 연관 관계를 나타낸다. 그리고 간략하게 각 서브 시스템의 개략적인 구조와 기능에 대해 설명한다.

3.2 System Organization

본 시스템은 가상현실 메타버스 플랫폼 중에 하나인 VRChat 이라는 플랫폼 내에서 구동되는 맵(월드)이다. 사용자와 관리자 모두 네트워크를 통해 VRChat 이라는 플랫폼에 접속하게 되고, 구현된 월드 내에서 시스템을 이용하게 된다. 플랫폼 내에서 호스트(최종 관리자)가 월드를 오픈하게 되면, 관리자와 사용자 모두 이 오픈 된 월드 내부로 들어간다. 월드는 스마트 캠퍼스라는 주제에 대응해 학교 오프라인 축제를 모티브로 개발되고, 실제 축제에서 사람들이 즐길 수 있는 부분들을 서브 시스템 혹은 기능으로 구현한다. 월드 내부의 요소들은 크게 관리자만 접근할 수 있는 축제 운영과 관련된 기능들, 그리고 일반 유저들이 사용할 수 있는 기능들로 나뉜다. 전자를 위해서 패스워드를 통해서 관리자 여부를 확인하는 시스템을 제작해, 이를 인증 과정으로 통과해야 특정 시스템에 접근할 수 있도록 설계하였다. 각각의 기능들을 담당하는 서브 시스템들은 상호 간 독립되어 있으며, 공통의 관리자 인증 시스템이 적용된다.

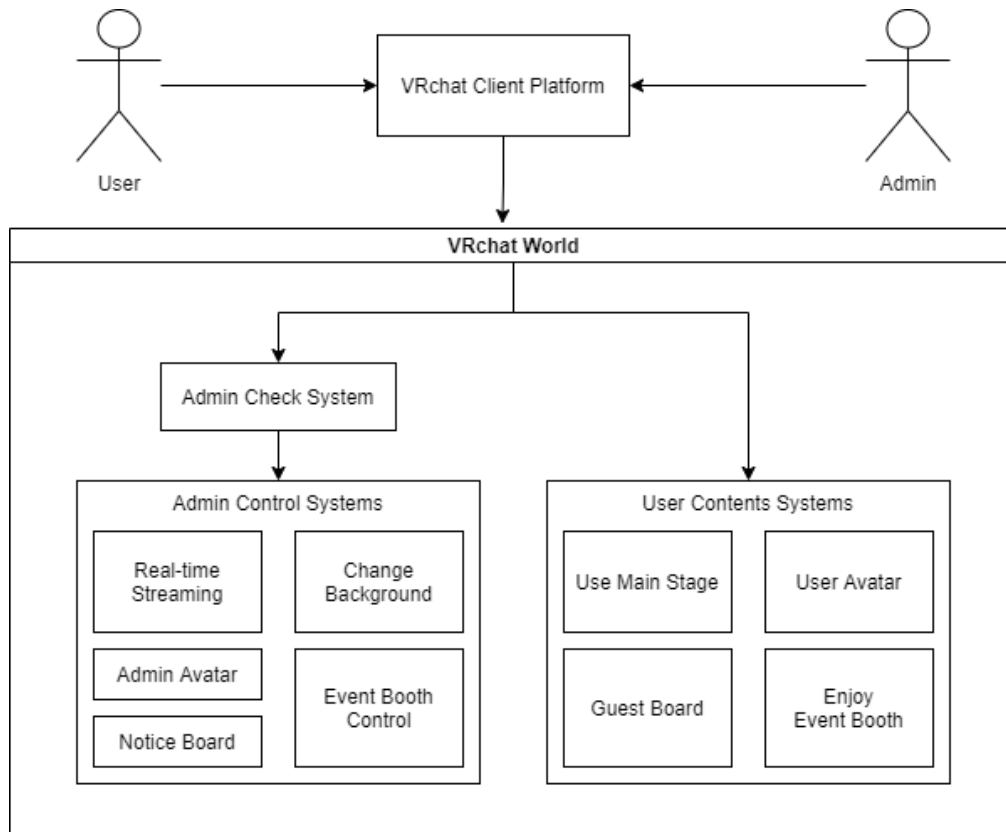


Diagram 1. Overall System Architecture

3.3 관리자 인증 시스템

관리자 인증 시스템은 본 서비스의 기능 중 실시간 스트리밍, 축제 분위기 전환, 게시판, 축제용 아바타 설정 서비스에 사용된다. 미리 설정된 비밀번호와 입력한 비밀번호가 일치하는지 확인하여 인증을 진행한다. 관리자 권한이 필요한 시스템을 먼저 식별하고, 비밀번호를 입력 받아 비교한 후, 비밀번호가 일치하면 해당 시스템을 정상적으로 activate 한다.

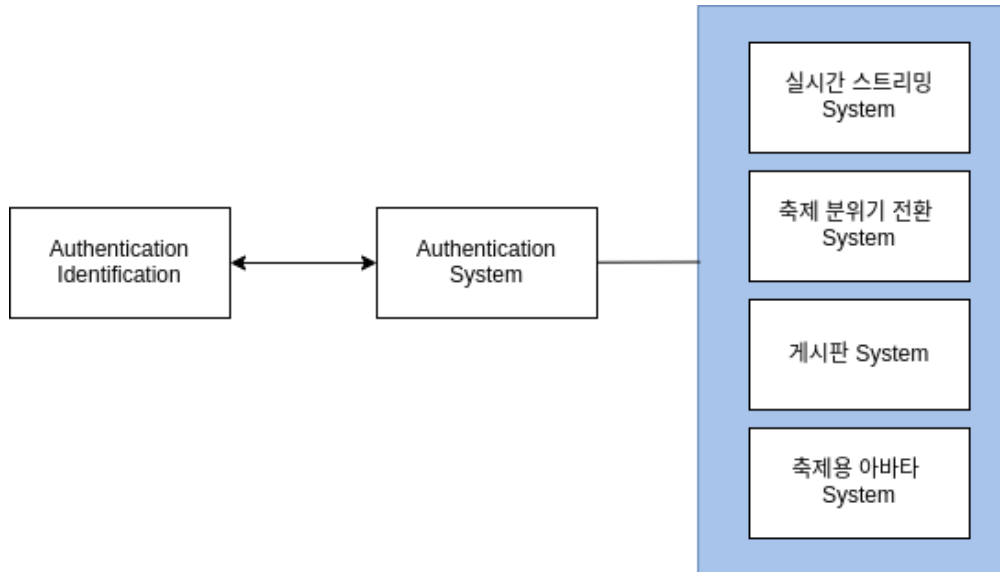


Diagram 2. Overall System Architecture – 관리자 인증

3.4 실시간 스트리밍 스크린

메인 미니 스크린에 영상을 업로드하고 관리자를 포함한 사용자들이 스크린을 통해 영상을 볼 수 있는 시스템이다. 스크린에 영상을 올리는 것은 관리자만 가능하기 때문에 패스워드를 통해 관리자를 확인한다. 관리자 확인을 마치면 비디오 파일 패스 또는 스트리밍 URL 을 통해 비디오를 스크린에 업로드한다.

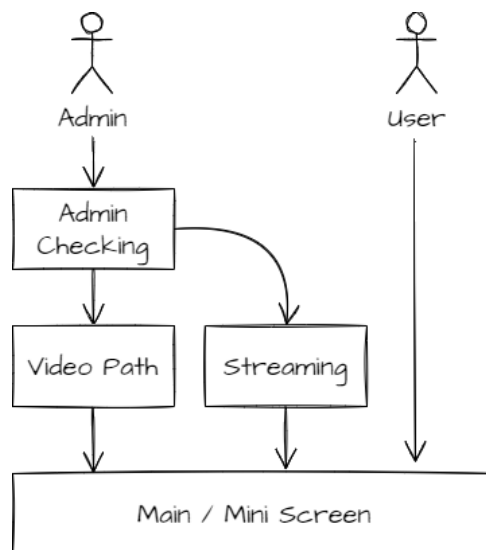


Diagram 3. Overall System Architecture – 실시간 스트리밍 스크린

3.5 축제 분위기 전환

관리자가 축제 분위기를 전환할 수 있는 시스템이다. 관리자 권한이 있는 사용자만이 분위기를 전환할 수 있기 때문에 패스워드를 통해 관리자를 확인한다. 관리자 권한이 확인되었을 때 먼저 모든 사용자에게 곧 분위기가 전환될 것임을 알리는 알림 문구가 뜨고 일정시간 후 분위기가 전환된다. 해당 시스템에서는 하늘 배경, 파티클 효과, 배경 음악을 상호 전환시킨다. 또한, 관리자는 특정 음악으로 배경음악을 변경할 수 있다.

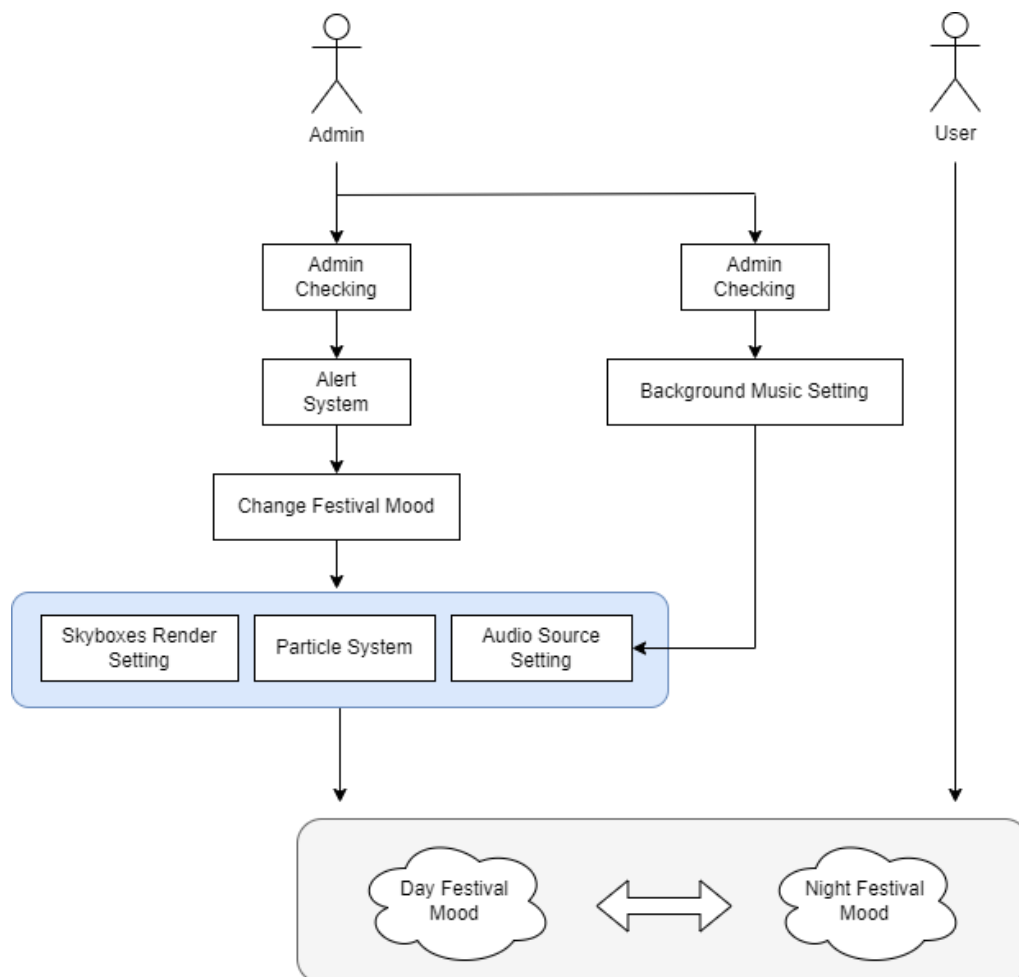


Diagram 4. Overall System Architecture – 축제 분위기 전환

3.6 게시판

게시판 시스템은 본 시스템에서 소통 수단 중 하나로 동작하며, Notice Board 와 Guest Bulletin Board 두 가지 종류로 나뉜다. Notice Board 는 공지사항, 홍보물 등이 게시되며 관리자만이 허가된 post 를 부착할 수 있다. 또한 '베스트 무대 투표'와 같은 사용자와 상호작용이 필요한 event post 도 게시될 수 있다. 반면 Guest Bulletin Board 는 짧은 글을 남길 수 있는 방명록 게시판으로, 모든 사용자가 읽고 쓸 수 있지만 부적절한 글은 사용자에게 의해 삭제될 수 있다. 두 종류의 게시판은 각각 1 개 이상 설치될 수 있다.

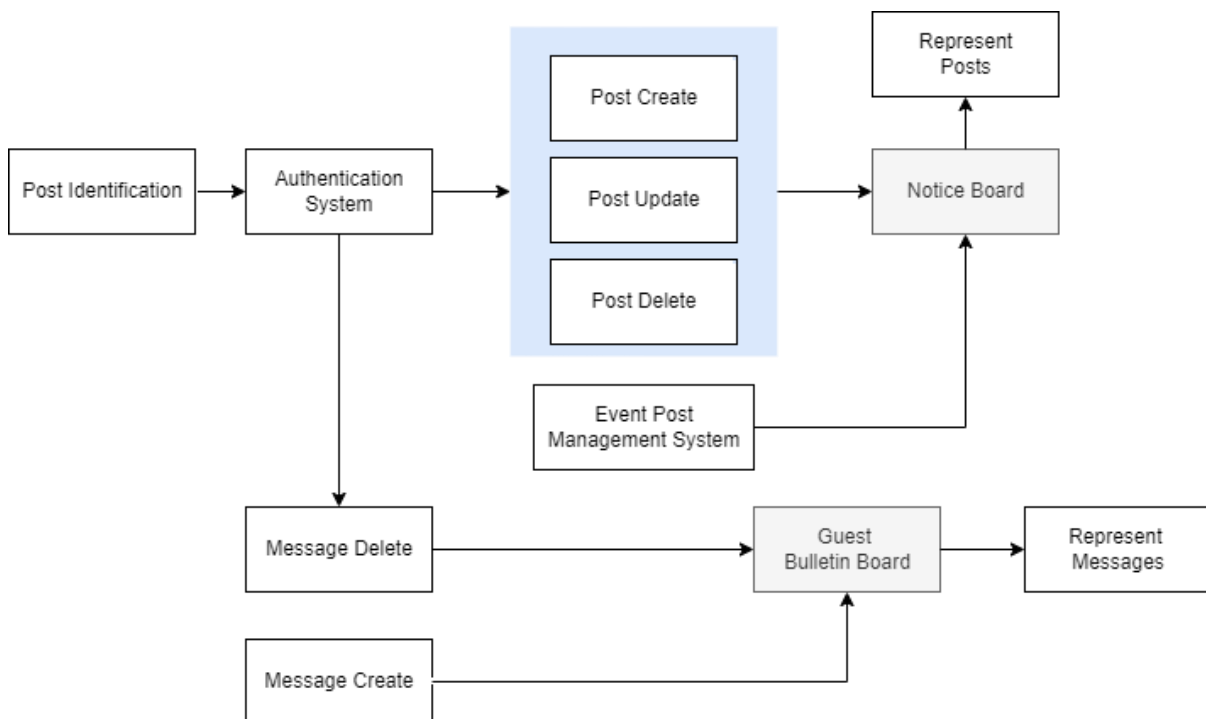


Diagram 5. Overall System Architecture – 게시판

3.7 이벤트 부스

이벤트 부스의 전체적인 구조는 위 그림과 같다. 이벤트 부스 섹션에는 소개팅 부스, 친목 도모 부스, 미니 게임 부스 등이 존재한다. 이 각각의 부스에는 관리자나 관리

역할을 하는 유저가 존재하는데, 이들은 이벤트 부스 입장 관리 인터페이스를 사용하여 이벤트 부스의 입장을 관리한다.

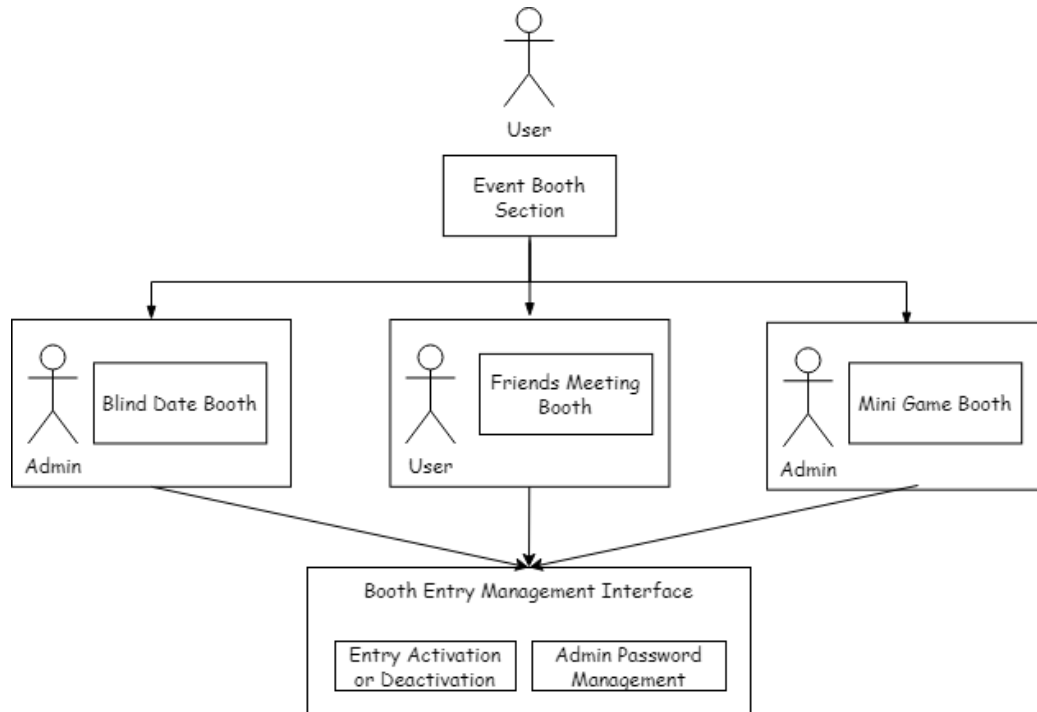


Diagram 6. Overall System Architecture – 이벤트 부스

3.8 축제용 아바타

축제용 아바타 시스템은 월드에 입장한 사용자들이 축제 참가를 위해 아바타를 변환하는 시스템이다. 축제에 입장한 사용자는 Avatar Pedestal 을 통해 자신의 아바타를 변환한다. Avatar Pedestal 은 프리뷰 사진을 통해 아바타의 모습을 사용자에게 미리 보여준다. 사용자는 아바타의 사진을 통해 자신이 사용할 아바타를 선택한다. 일반 이용자용 아바타는 축제에 입장한 모든 사람이 사용할 수 있다. 반면에 관리자용 아바타는 관리자만이 사용할 수 있다. 관리자용 아바타의 Avatar Pedestal 은 패스워드를 통해 관리자를 식별한 후 활성화된다. 아바타가 변환될 때, 텍스트 메시지를 통해 사용자 및 관리자의 입장을 환영한다.

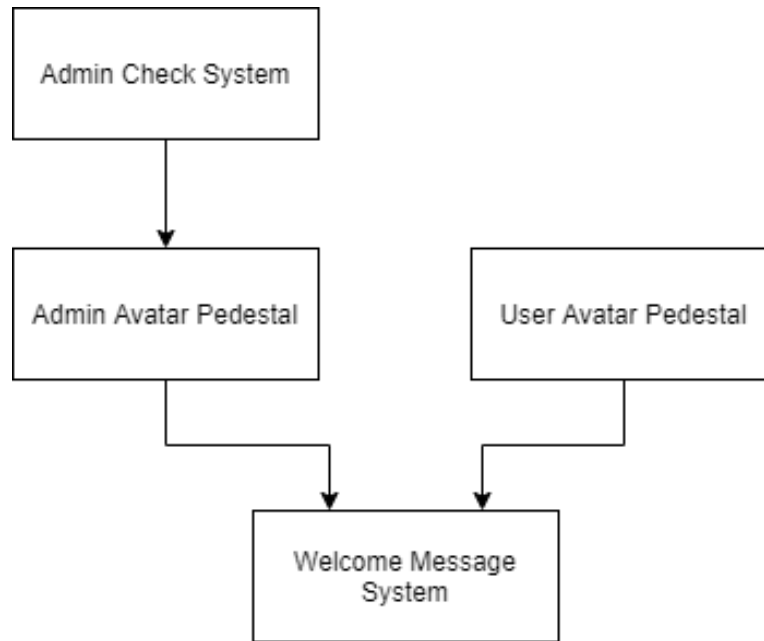


Diagram 7. Overall System Architecture – 아바타

4 System Architecture : 관리자 인증

4.1 Objectives

관리자 인증 시스템은 본 서비스의 기능 중 관리자 권한이 필요한 경우에 사용되며, 관리자는 비밀 번호를 통해 인증된다. 이 챕터에서는 관리자 인증 시스템의 Class Diagram, Sequence Diagram 을 나타낸다.

4.2 Overall Architecture

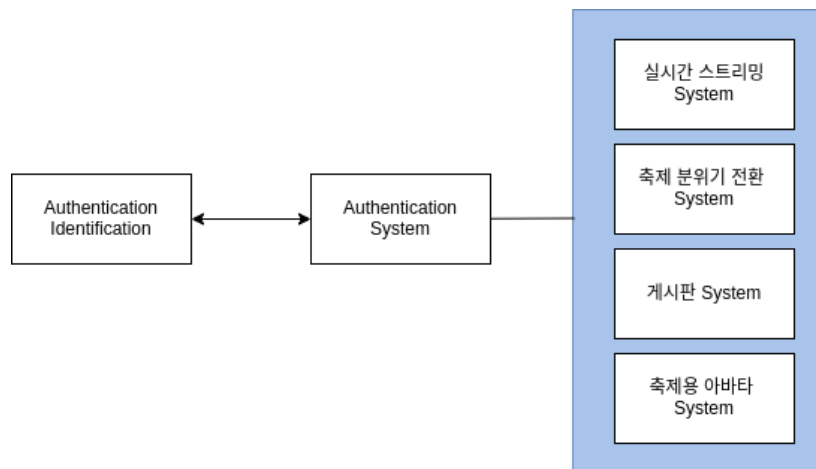


Diagram 8. Block Diagram – 관리자 인증

4.3 Detailed Architecture

4.3.A Class Diagram

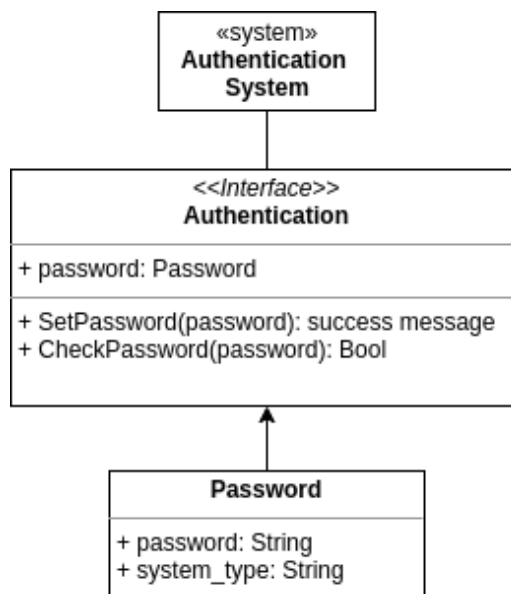


Diagram 9. Class Diagram – 관리자 인증

String 으로 비밀번호를 구성하며, 관리자는 비밀번호를 설정할 수 있다. SetPassword() 함수를 통해 비밀번호를 최초 설정 혹은 재설정할 수 있다. CheckPassword()는 파라미터로 입력된 패스워드가 기존에 설정된 패스워드와 일치하는지 여부를 반환한다.

4.3.B Sequence Diagram

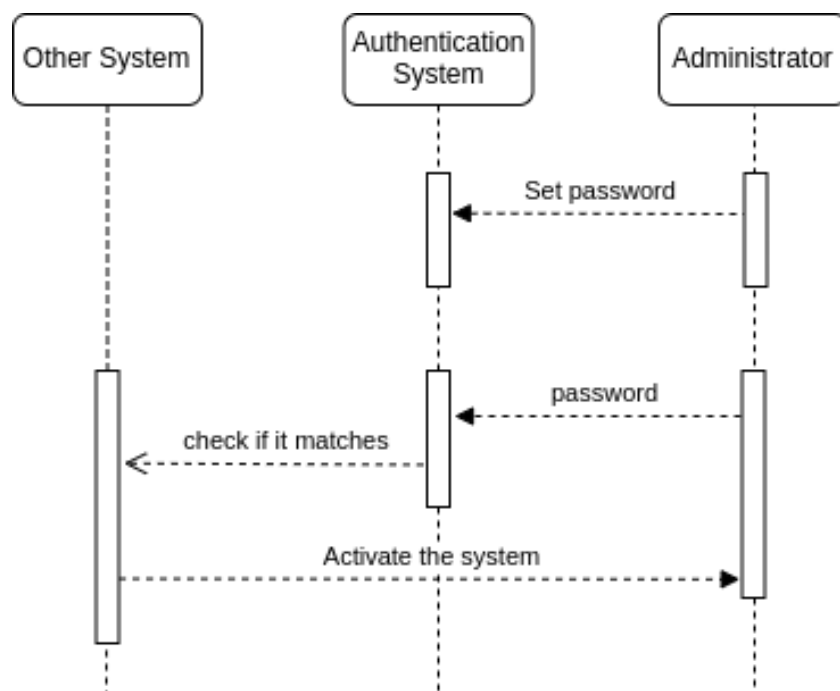


Diagram 10. Sequence Diagram – 관리자 인증

Other System 은 위의 Overall Architecture 에서 나타냈듯이, 관리자 인증 시스템이 필요한 모든 시스템들을 말한다. 관리자는 패스워드를 설정할 수 있고, 필요한 경우 패스워드를 입력하면 시스템이 일치 여부를 확인하여 사용하고자 하는 시스템에 반환 값을 전달한다. 반환 값이 true 인 경우 해당 시스템을 활성화한다.

5 System Architecture : 실시간 스트리밍 스크린

5.1 Objectives

실시간 스트리밍 스크린을 통해 녹화 영상 패스나 실시간 스트리밍 링크를 통해 스크린에 영상을 띄우기 위한 시스템이다. 오직 관리자만 영상 콘텐츠를 변경할 수 있다. 관리자를 포함한 사용자들은 메인/미니 스크린을 통해서 영상 콘텐츠를 시청할 수 있다. 이 시스템들의 기능과 설계를 설명하고, Class Diagram 와 Sequence Diagram 을 통해 구조를 표현한다.

5.2 Overall Architecture

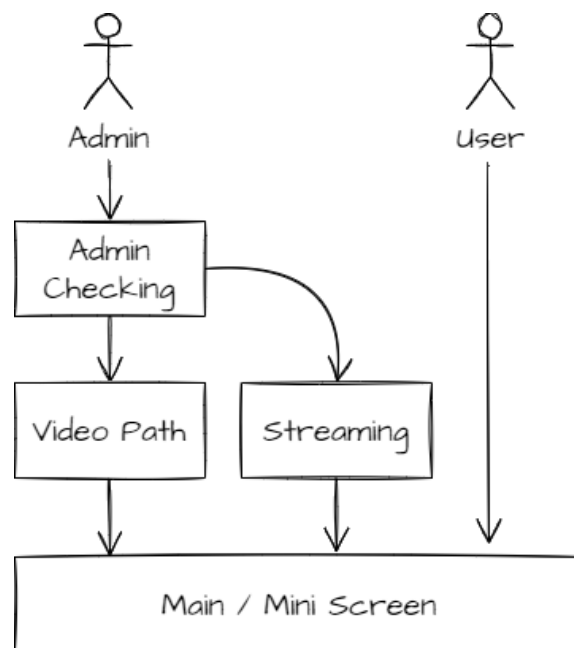


Diagram 11. Block Diagram - 실시간 스트리밍 스크린

5.3 Detailed Architecture

5.3.A Class Diagram

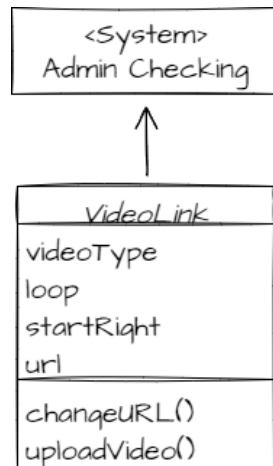


Diagram 12. Class Diagram - 실시간 스트리밍 스크린

5.3.B Sequence Diagram

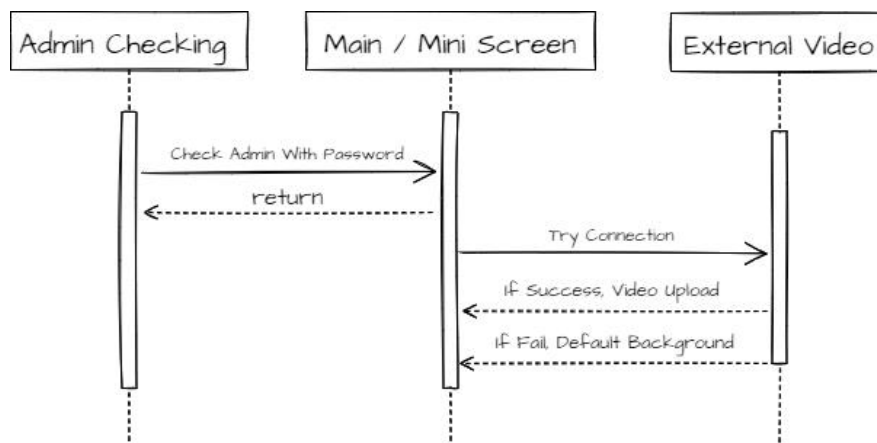


Diagram 13. Sequence Diagram - 실시간 스트리밍 스크린

5.3.C State Diagram

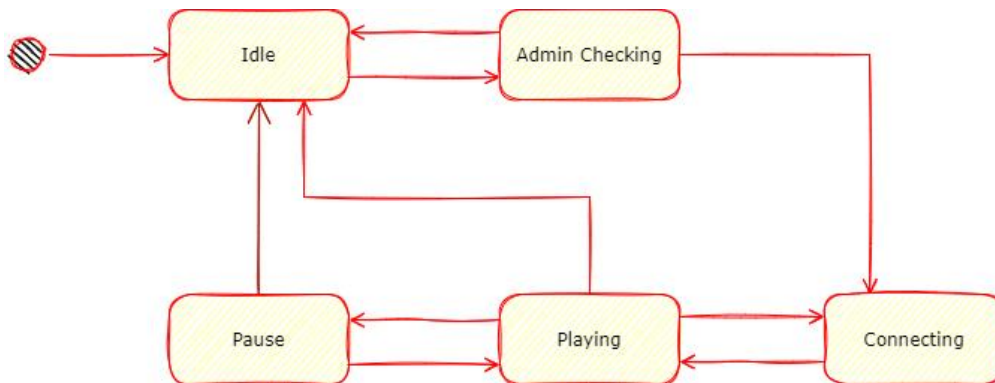


Diagram 14. State Diagram - 실시간 스트리밍 스크린

6 System Architecture : 축제 분위기 전환

6.1 Objectives

관리자 권한을 가진 사용자가 축제 분위기를 낮/밤에 따라 전환시킬 수 있는 시스템이다. 오직 관리자만이 전환할 수 있다. 해당 기능에 따라 모든 사용자는 모다 다양한 분위기의 축제를 즐길 수 있다. 이 시스템의 기능과 세부 특성들을 아래 Class Diagram 과 Sequence diagram 을 통해 기술한다.

6.2 Overall Architecture

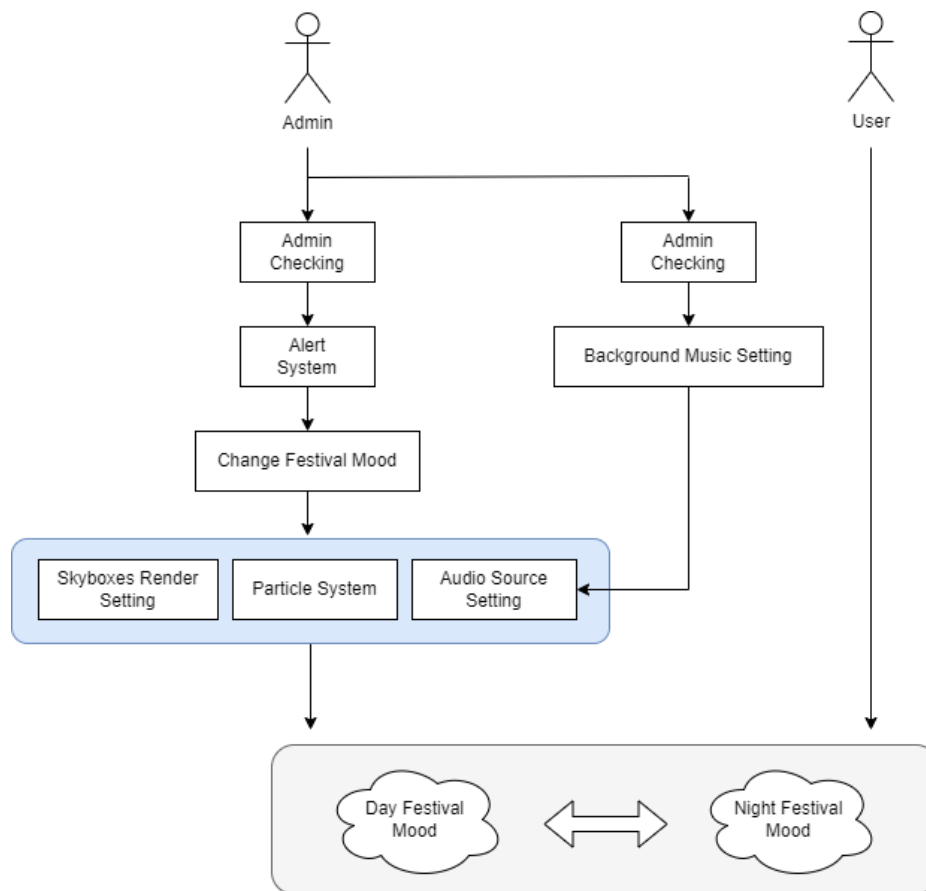


Diagram 15. Block Diagram - 축제 분위기 전환

6.3 Detailed Architecture

6.3.A Class Diagram

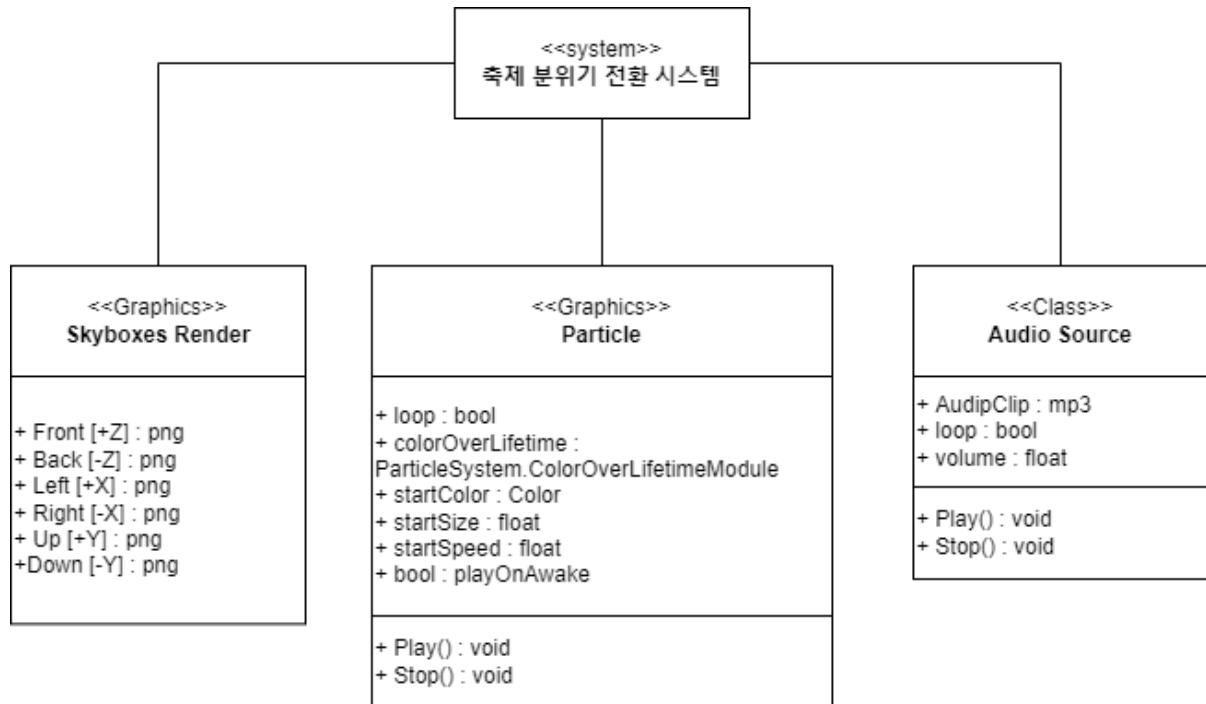


Diagram 16. Class Diagram – 축제 분위기 전환

6.3.B Sequence Diagram

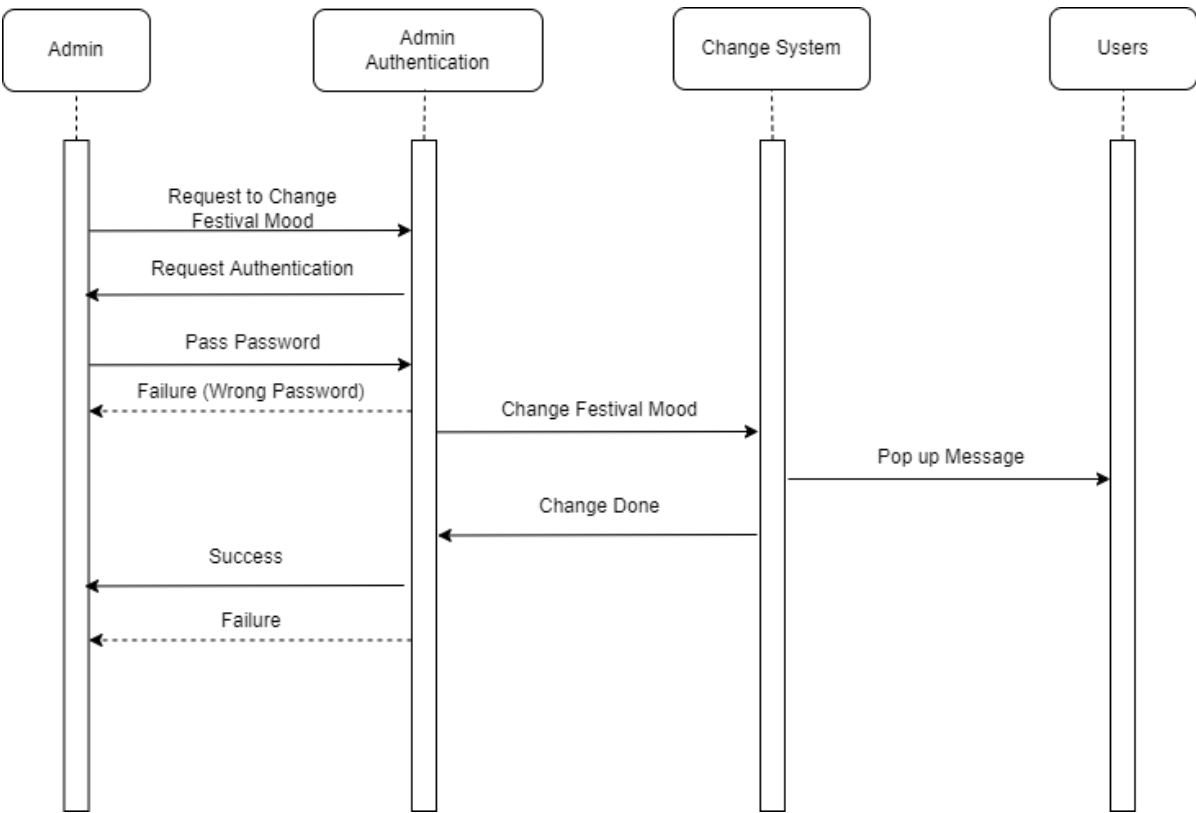


Diagram 17. State Diagram - 축제 분위기 전환

6.3.C State Diagram

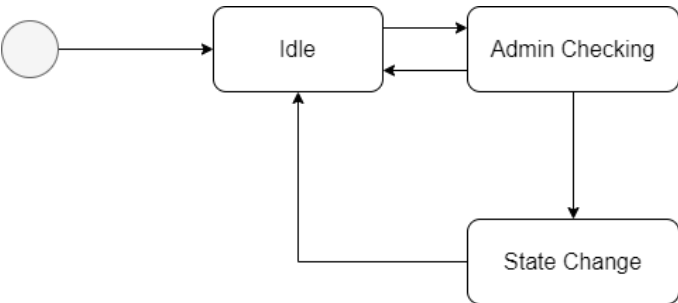


Diagram 18. State Diagram - 축제 분위기 전환

7 System Architecture : 게시판

7.1 Objectives

Notice Board 에는 관리자가 공지 사항, 이벤트 게시물을 게시할 수 있으며 Guest Bulletin Board 에는 모든 사용자가 방명록을 읽고 쓸 수 있다. 이 chapter에서는 각 게시판이 어떤 특성들을 가지고 어떤 순서로 동작하는지를 Class Diagram 과 Sequence Diagram 을 통해 기술한다.

7.2 Overall Architecture

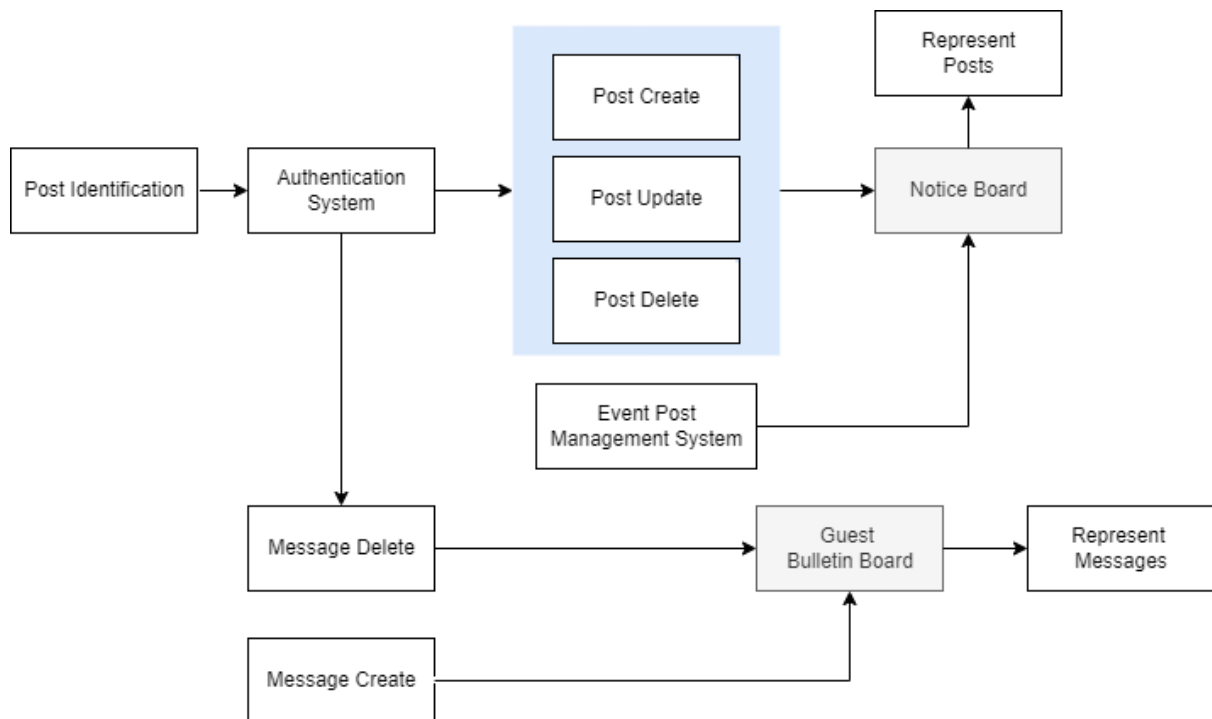


Diagram 19. Block Diagram – 게시판

7.3 Detailed Architecture

7.3.A Class Diagram

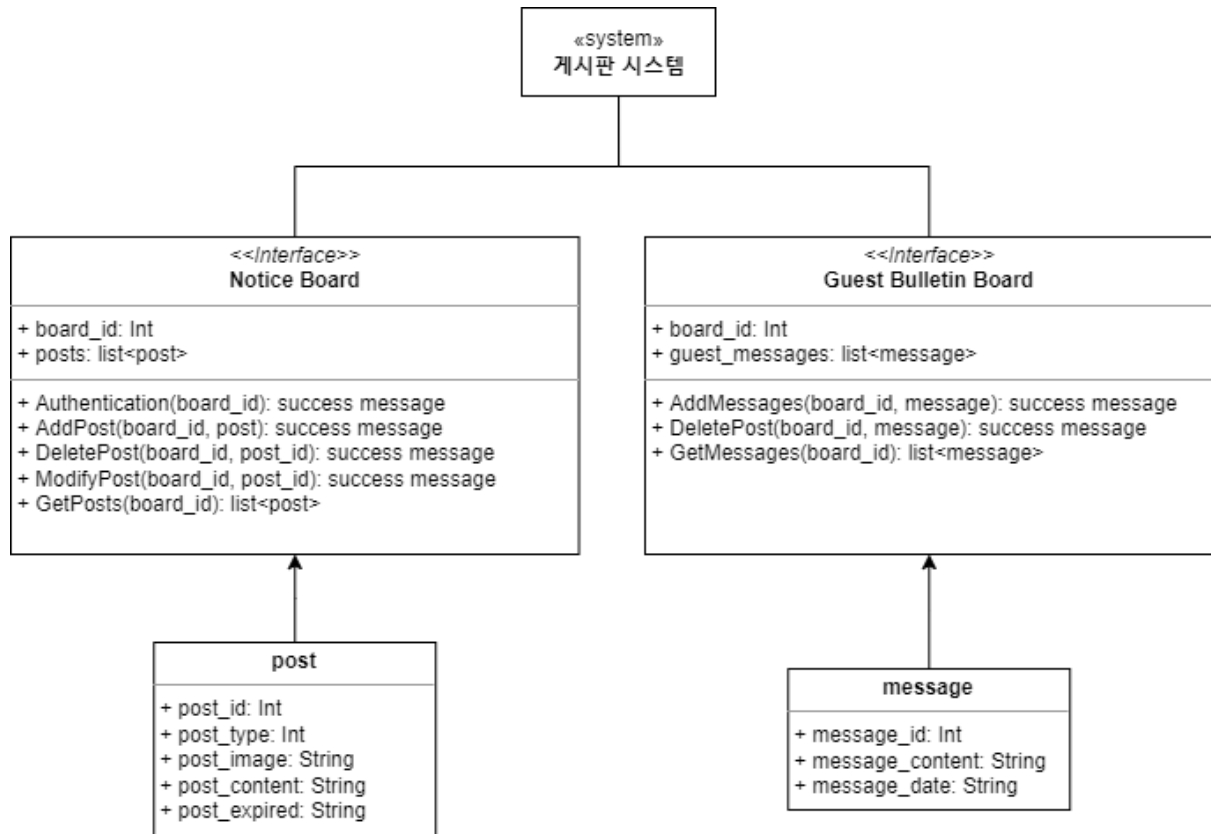


Diagram 20. Class Diagram – 게시판

7.3.B Sequence Diagram

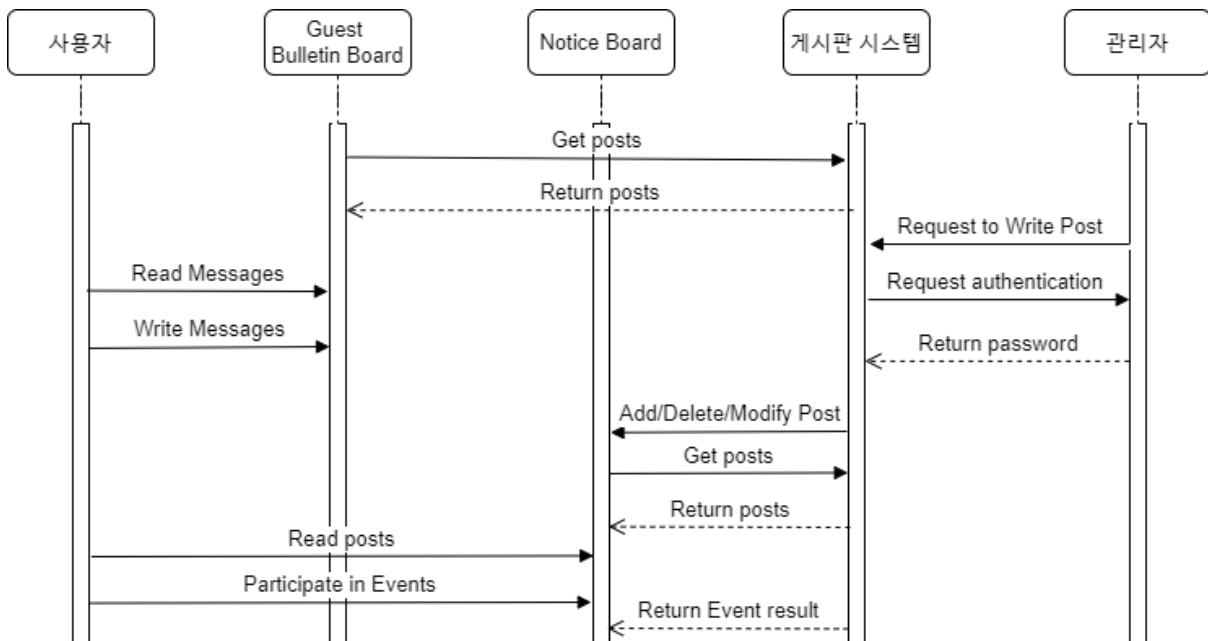


Diagram 21. Sequence Diagram – 게시판

8 System Architecture : 이벤트 부스

8.1 Objectives

이벤트 부스는 사용자들이 축제 중에 즐길 수 있는 이벤트들이 발생하는 섹션이다. 이 챕터에서는 이벤트 부스의 sub-system 들에 대해 설명하고 상호작용 과정에 대해 설명한다.

8.2 Overall Architecture

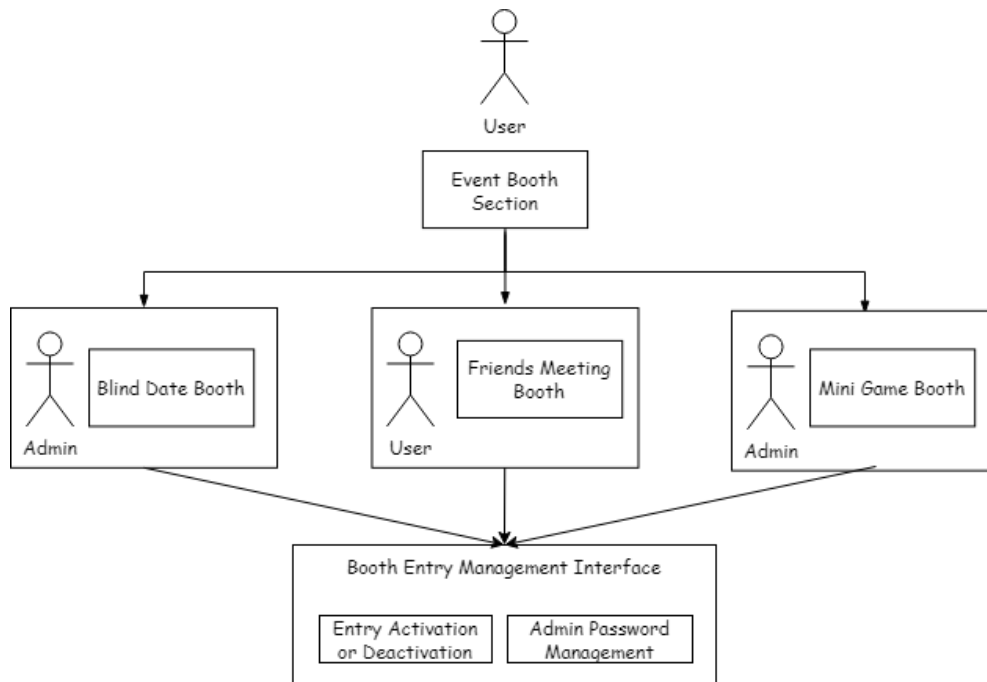


Diagram 22. Overall System Architecture – 이벤트 부스

8.3 Detailed Architecture : 블라인드 소개팅 부스

블라인드 소개팅 부스는 2 명의 유저들이 아바타를 사용하여 소개팅을 할 수 있는 부스이다. 소개팅 참가자들의 입장을 관리하는 관리자가 존재한다.

8.3.A Class Diagram

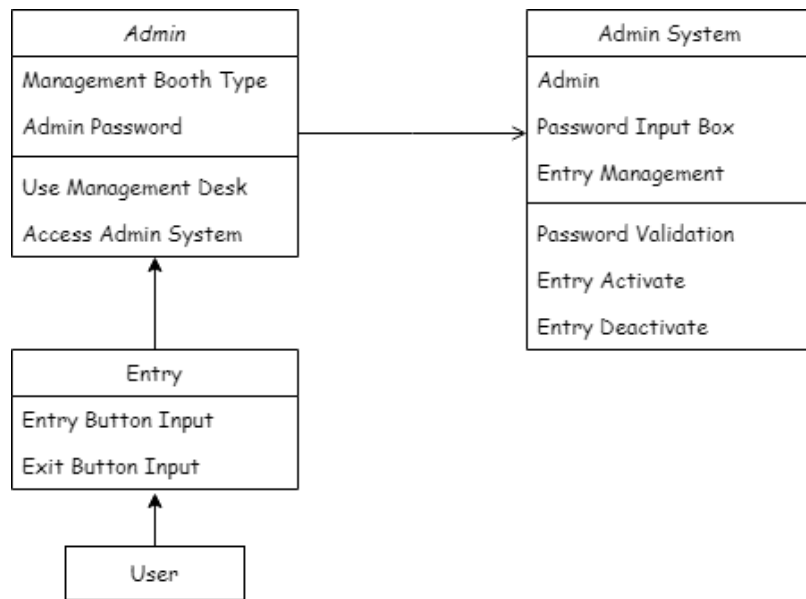


Diagram 23. Class Diagram – 이벤트 부스 : 블라인드 소개팅

Table 1. Class Description – 이벤트 부스 : 블라인드 소개팅

Class	Description
Admin System	관리자가 관리시스템을 총괄할 수 있도록 하는 sub-system 이다. 관리자는 해당 관리시스템에 접근할 수 있는 비밀번호를 사전에 설정하고 비밀번호 검증을 통해 관리시스템을 사용할 수 있다.
Admin	관리 권한이 있는 관리자이다. 블라인드 소개팅과 미니게임 부스에는 관리자가 필요하므로 관리자에게는 관리하는 부스 타입이 있어야 한다. 또한 관리자는 관리 시스템에 접근할 수 있는 비밀번호를 알고 있어야 한다.

Entry	유저가 입장을 하기 위해 사용하는 인터페이스 클래스다. 입장, 퇴장 버튼 인풋을 받는다.
User	소개팅 부스를 사용하고자 하는 유저이다. 한 부스에 최대 2 명의 유저가 들어갈 수 있다.

8.3.B Sequence Diagram

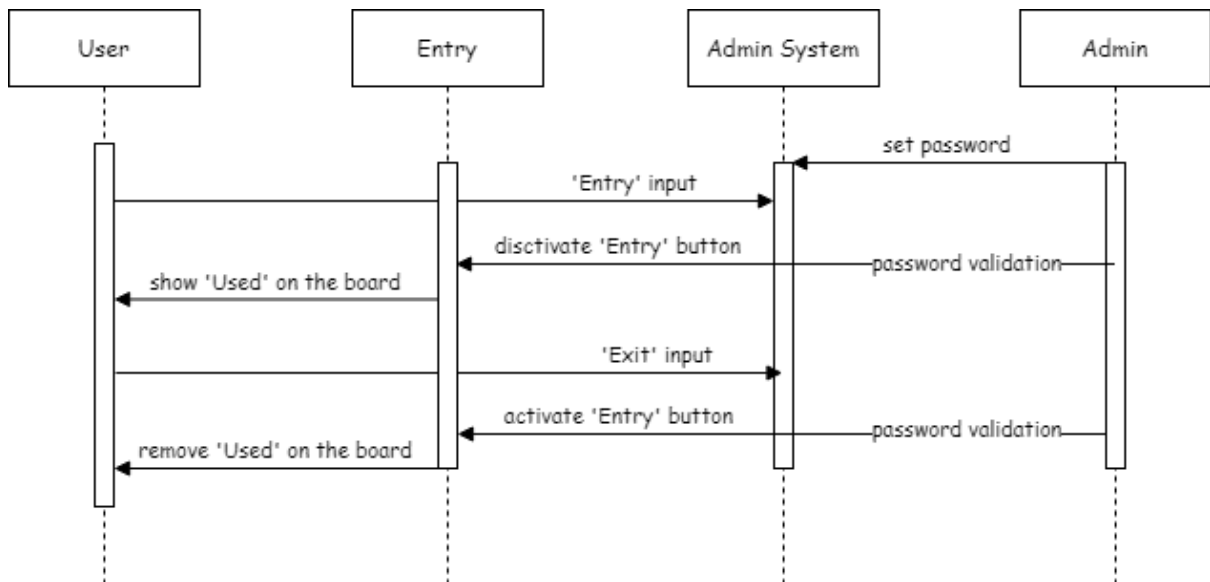


Diagram 24. Sequence Diagram – 이벤트 부스 : 블라인드 소개팅

8.4 Detailed Architecture : 친목도모 부스

친목도모 부스는 이용자들이 모여서 함께 친목을 도모할 수 있는 부스이다. 친목도모 부스에는 관리자가 따로 없으며, 이용자가 직접 관리 시스템에 접근할 수 있다. 혼선을 방지하기 위해 한 이용자만이 관리 시스템에 접근 가능하고, 이용자가 더 이상 친목도모 부스를 사용하지 않을 시에, 관리 시스템 접근 권한을 직접 해지한다.

8.4.A Class Diagram

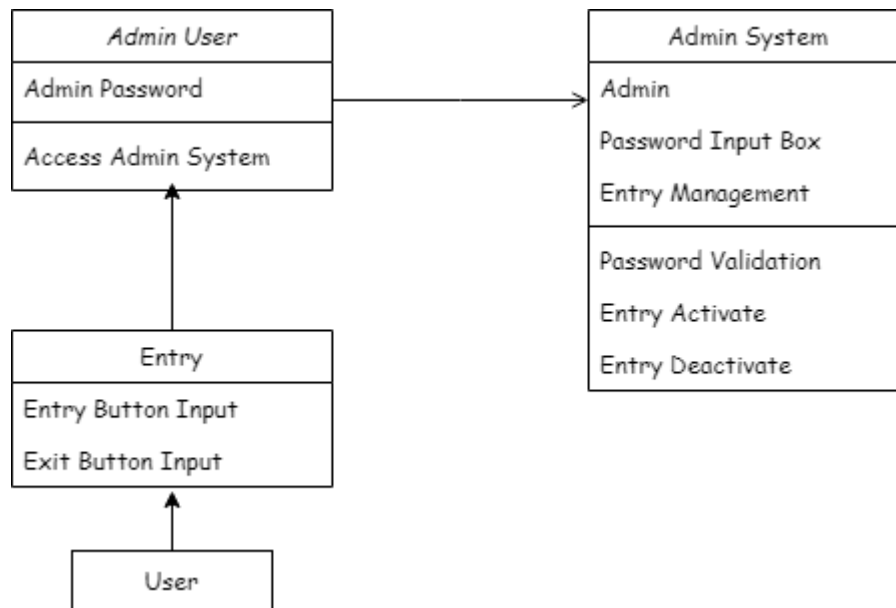


Diagram 25. Class Diagram – 이벤트 부스 : 친목도모 부스

Table 2. Class Description – 이벤트 부스 : 친목도모 부스

Class	Description
Admin System	소개팅 부스에서의 admin system 과 같다.
Admin User	친목도모 부스에서의 관리자는 축제 유저이다. 유저는 만약 해당 부스의 관리 시스템의 관리자가 비어 있을 경우, 직접 친목도모 부스 관리 시스템의 비밀번호를 설정할 수 있다. 비밀번호를 설정한 유저는 해당 부스 관리 시스템에 접근 가능하다.
Entry	유저가 입장을 하기 위해 사용하는 인터페이스 클래스다. 입장, 퇴장 버튼 인풋을 받는다.

User	친목도모 부스를 사용하고자 하는 유저이다. 한 부스에 들어갈 수 있는 유저의 수에는 제한이 없다.
------	--

8.4.B Sequence Diagram

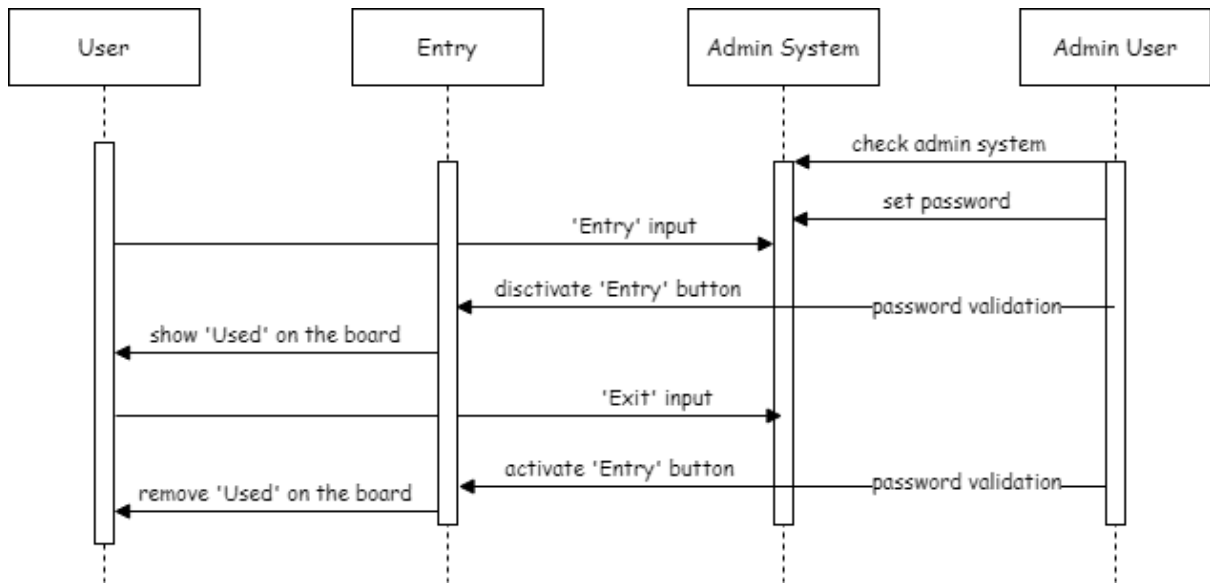


Diagram 26. Sequence Diagram – 이벤트 부스 : 친목도모 부스

8.5 Detailed Architecture : 미니게임 부스

미니게임 부스는 축제 이용자들이 미니게임을 즐길 수 있는 부스이다. 한 미니게임 부스에는 한 명의 관리자가 존재하며 유저의 입장을 관리한다.

8.5.A Class Diagram

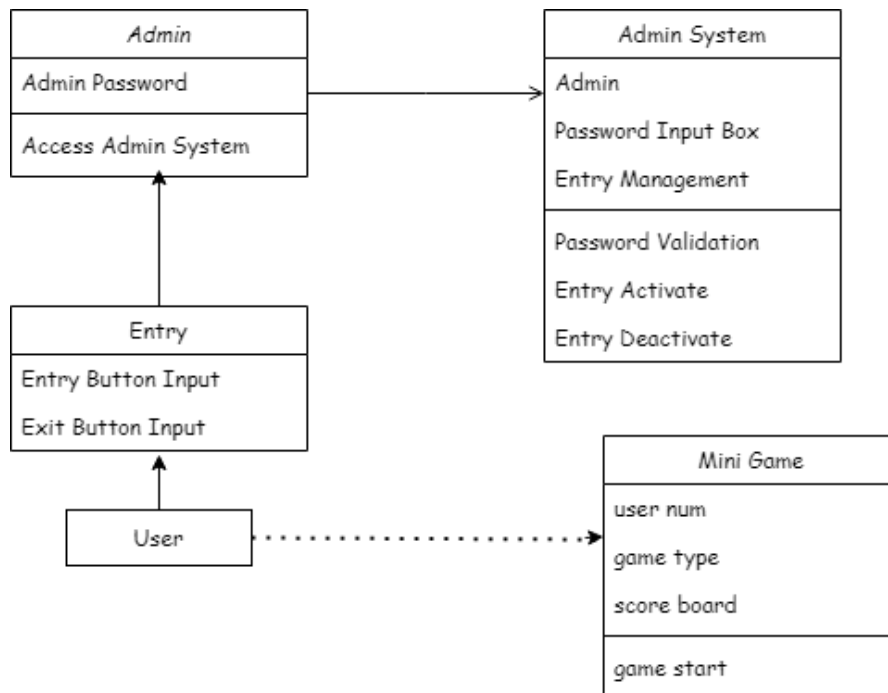


Diagram 27. Class Diagram – 이벤트 부스 : 미니게임 부스

Table 3. Class Description – 이벤트 부스 : 미니게임 부스

Class	Description
Mini Game	해당 부스에서 발생하는 미니게임이다. 사용자들이 게임을 위해 의사소통을 할 수 있는 인터페이스를 제공한다.

8.5.B Sequence Diagram

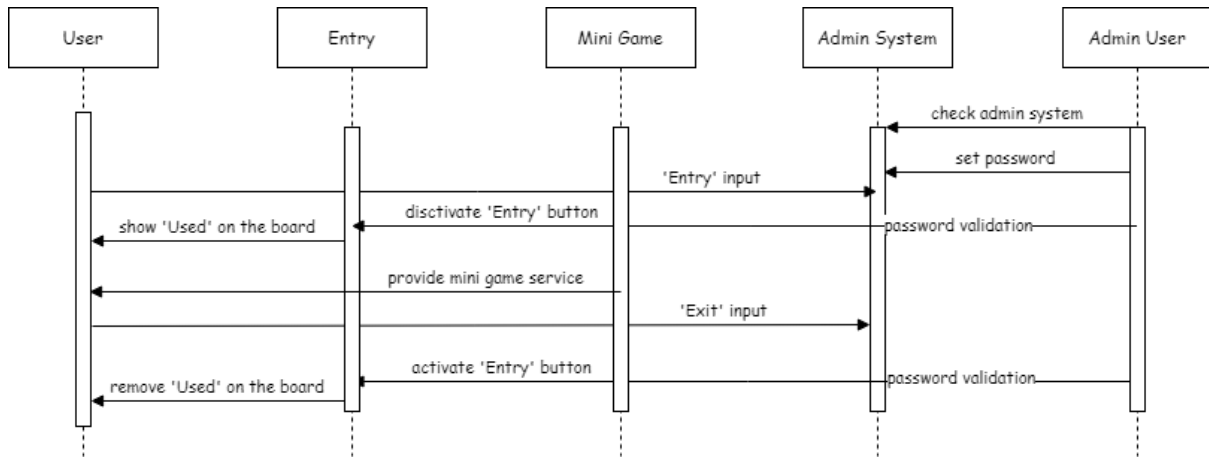


Diagram 28. Sequence Diagram – 이벤트 부스 : 미니게임 부스

9 System Architecture : 축제용 아바타

9.1 Objectives

축제용 아바타 시스템은 일반 사용자 및 관리자가 축제 참가를 위해 아바타를 변환하는 시스템이다. 관리자용 아바타는 관리자만이 사용할 수 있다. 관리자용 아바타의 Avatar Pedestal 은 패스워드를 통해 관리자를 식별한 후 활성화된다. 아바타가 변환될 때, 텍스트 메시지를 통해 사용자 및 관리자의 입장을 환영한다.

9.2 Overall Architecture

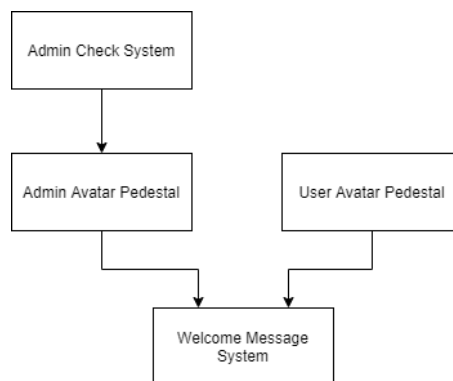


Diagram 29. Block Diagram – 아바타

9.3 Detailed Architecture

9.3.A Class Diagram

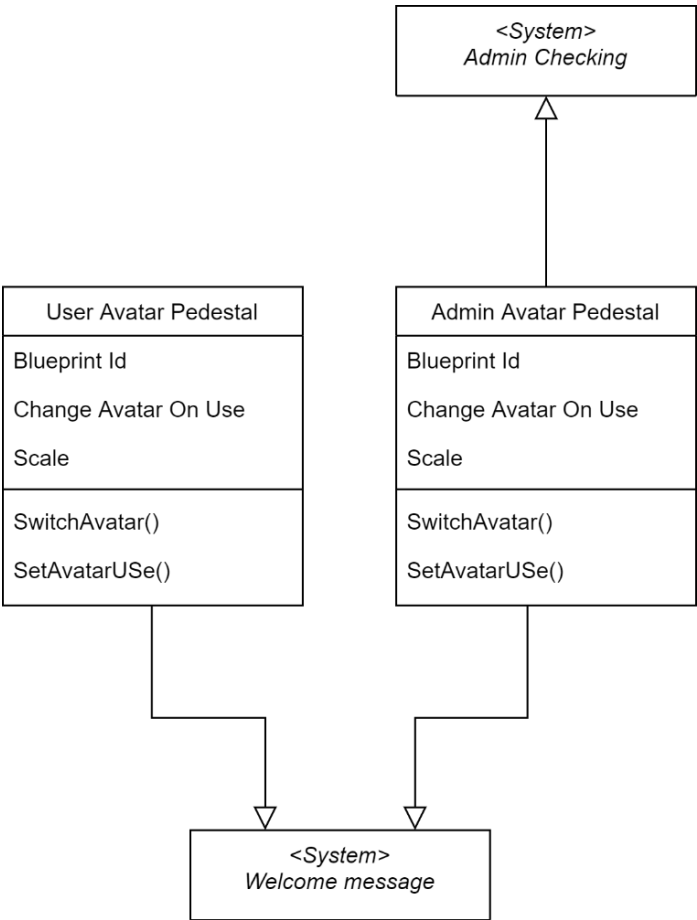


Diagram 30. Class Diagram – 아바타

9.3.B Sequence Diagram

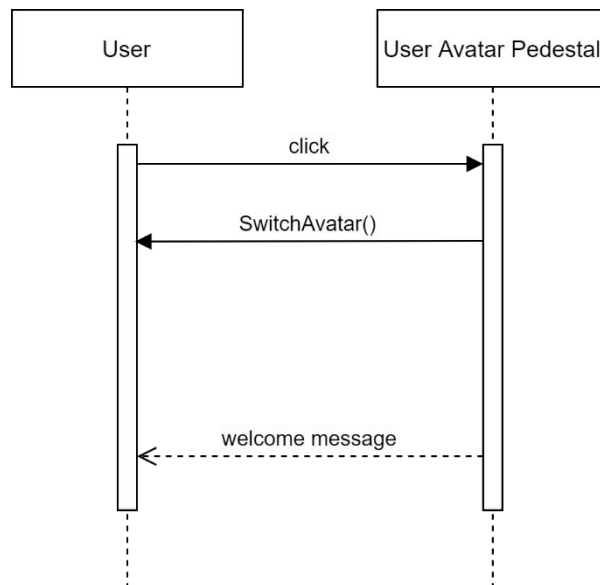


Diagram 31. Sequence Diagram – 유저 아바타

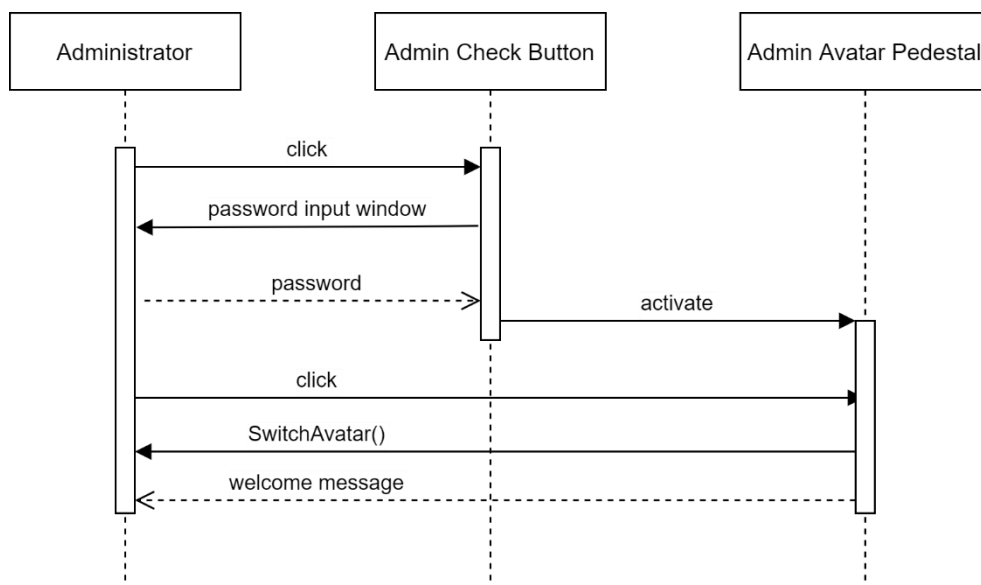


Diagram 32. Sequence Diagram – 관리자 아바타

10 비디오-유튜브 연결 Design

10.1 Objectives

Streaming Live Design 은 요구사항 명세서에서 기술한 요구사항을 바탕으로 작성하였다. 요구사항을 바탕으로 ER Diagram 을 작성하고, 이를 통해 Relational Schema 와 SQL DDL 을 작성한다.

10.2 ER Diagram

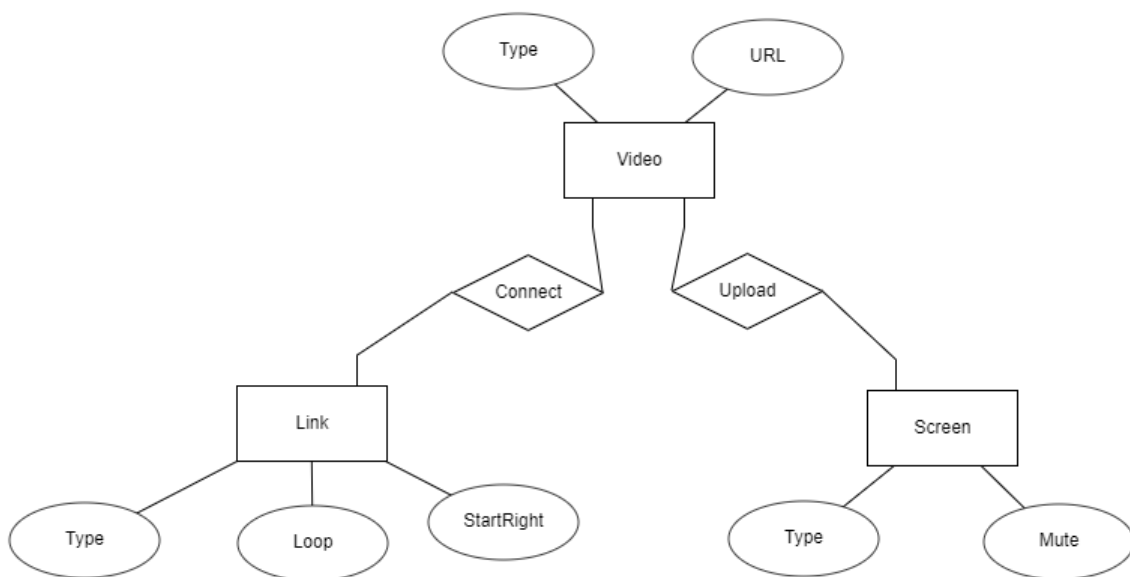


Diagram 33. ER Diagram – 비디오-유튜브 연결

10.2.A Entities

10.2.A.1 Link

Link 는 영상을 업로드 방식 정보를 나타낸다. Type, Loop, StartRight 속성을 가지고 있다.

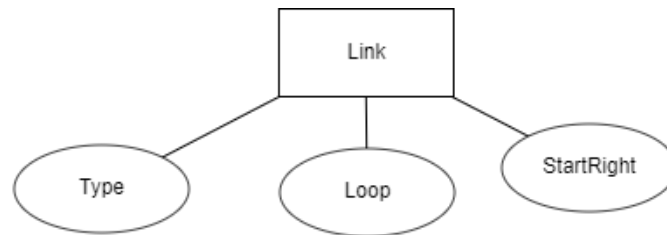


Diagram 34. ER Diagram - Link

10.2.A.2 Video

Video 는 영상 정보를 나타낸다. Type, URL 속성을 가지고 있다.

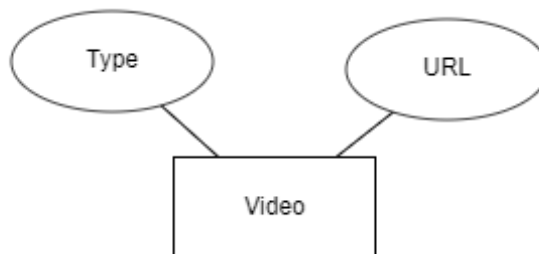


Diagram 35. ER Diagram – Video

10.2.A.3 Screen

Screen 은 스크린 정보를 나타낸다. Type, Mute 속성을 가지고 있다.

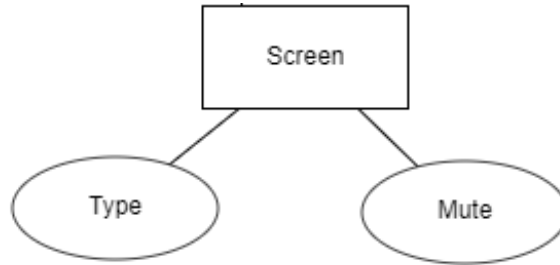


Diagram 36. ER Diagram – Screen

10.2.B Relation

10.2.B.1 Connect

Connect Relationship 은 관리자가 입력한 비디오 타입에 따라 연결한다. 한 영상당 여러 개의 링크 방식을 가질 수 있다.

10.2.B.2 Upload

Upload Relationship 은 관리자가 선택한 비디오를 스크린에 띄운다. 한 영상당 여러 스크린에 띄워질 수 있다.

10.3 Relational Schema

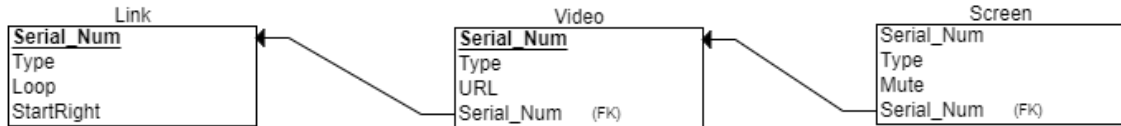


Diagram 37. Relational Schema

11 Testing Plan

11.1 Objectives

이 챕터에서는 요구사항 명세서에서 기술한, 사용자 및 관리자 시나리오를 바탕으로 한 전체 시스템이 구상한대로 잘 실행되는지 확인한다. 또한 발생할 수 있는 오류와 결함을 감지하고 이를 발견해 수정함으로써 보다 완벽한 작동과 안정성을 갖추어 출시한다는 중요점을 충족하기 위해 수행한다. 관련 테스트를 수행하기 위한 내용을 기술하고 개발 테스트, 릴리스 테스트 및 사용자 테스트로 나누어 테스트 정책을 수립한다. 그리고 구체적인 유저 테스트 케이스를 만든다.

11.2 Testing Policy

11.2.A Development Testing

개발 테스트는 소프트웨어 개발의 잠재적 위험을 줄이고 시간과 비용을 절약하기 위해 광범위한 결함 예방 및 탐지 전략의 동기화된 적용을 주로 수행한다. 이 단계에서는

소프트웨어가 충분한 테스트를 거치지 않았기 때문에 불안정할 수 있으며 구성 요소들이 서로 충돌할 수 있다. 따라서 이 단계에서는 정적 코드 분석, 데이터 흐름 분석, 피어 코드 검토, 유닛 테스트가 수행되어야 한다.

일부 성능과 보안, 안정성과 관련된 부분에 대한 검증은 특정 메타버스 플랫폼(VRChat) 상에서의 월드 개발이라는 프로젝트 특성을 고려해 테스트 정책에 반영한다. 각각의 서버 시스템과 기능들이 요구사항 명세서에서 작성한 것과 같이 정상적으로 작동하는지를 테스트하는 과정 이후에 통합해서 전체 시스템 테스트를 수행할 예정이다. 이러한 과정을 통해 팀 프로젝트 특성에 맞추어 다음과 같은 세 가지 비기능적 요소를 테스트하는데 집중한다.

11.2.A.1 Usability

유저들의 사용편의성과 관련된 부분으로, 직관적인 UI 구성과 현실 축제에서 행하던 여러가지 모습들을 반영함으로써, 메타버스에 익숙하지 않은 사람들도 쉽게 적응하고 사용하고 즐길 수 있도록 월드를 제작해야 한다.

11.2.A.2 Security

사용자의 정보 및 시스템의 확보는 고려해야 하는 중요한 문제이며, 정보의 가치와 무관하게 허락되지 않은 이용자가 정보에 접근하는 것을 방지해야 한다. 게시판과 스크린, 이벤트 부스 등에서 당사자가 원하지 않는 정보가 유출되거나 공개될 상황이 있는지 테스트한다.

11.2.A.3 Reliability

VRChat 이라는 플랫폼에 있어서 서버나 네트워크, 트래픽 관련 이슈들은 없었는지 체크하고 월드를 제작한 후에 자체적 테스트를 거쳐 원활한 기능 작용이 실제로 가능한지 확인해 보아야 한다. 스트리밍 사이트를 제공하는 업체들의 속도나 끊김 관련 체크가 필요하고, 다른 사용자가 많이 몰리는 VRChat 내부의 다른 월드를 참고로 삼아

현황을 파악한다. 즉 기능적인 요소들이 개별적으로 정상적으로 작동하는지 뿐만 아니라 축제 월드가 굉장히 활성화된 전체적인 상황을 가정하여 테스트를 시행해서 신뢰성을 확보한다.

11.2.B Release Testing

소프트웨어 개발 프로젝트의 중요한 부분 중 하나는 시장과 고객에게 제품을 출시하는 것이다. 기술적으로 좋은 성능을 가진 소프트웨어라도 잘못된 릴리스 방식으로 인해 성공하지 못할 수도 있다. 따라서 제품과 시장의 연결성을 높이기 위해 릴리즈 테스트가 불가피하다. 릴리스 테스트는 소프트웨어/애플리케이션의 최신 버전 /빌드를 테스트하여 결함이 없고 문제가 없는지, 의도한 대로 작동하는지 확인하는 프로세스이다. 최종적으로 완성된 시스템을 배포하기 전에 실시하는 테스트이며, 시스템이 사용자의 요구사항을 모두 충족하는지, 서비스를 개발의도에 맞게 제공하는지, 정상적으로 기능이 작동하는지 등을 확인한다. 소프트웨어 릴리스 수명 주기에 기반하여 테스트는 일반적으로 소프트웨어의 기본 구현이 완료되는 '알파' 버전부터 시작된다. 알파 버전에서 개발 테스트를 시작하고, 사용자 및 릴리스 테스트를 포함한 추가 테스트를 위해 베타 버전을 완성하여 실제 사용자들로부터 피드백을 받아 이를 보완해 완성된 소프트웨어를 최종 완성할 예정이다.

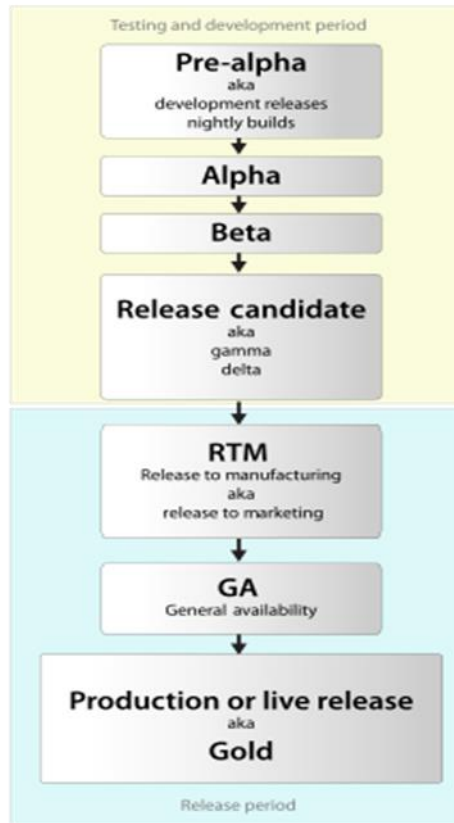


Figure 9. Software Release Life Cycle

11.2.C User Testing

이 단계는 사용자의 실제 사용 환경에서 시스템을 테스트한다. 실제 사용자들을 통해 소프트웨어가 직관적인지, 사용하기 쉬운지 등을 평가하고 소프트웨어 출시의 가능 여부를 판단하게 된다. 사용자 시험은 가용성 시험의 개념으로 설명될 수 있다. 필요한 사용자 테스트를 진행할 수 있는 시나리오와 현실적인 상황을 설정하여 진행할 예정이다.

11.2.D Testing Case

여기서 제안되는 일부 테스트 케이스 예시들은 development testing 과정에서, 그 중에서도 특히 기능적인 요소들을 검증하는 목적으로 쓰인다. 관리자 시스템이 각 서브시스템과 결합해서 정상적으로 작용하는지 여부, 인증 이후에 관리자 기능들이

정상적으로 동작하는지, 일반 사용자들이 허락된 기능을 제약 없이 정상적으로 사용할 수 있는지 등의 케이스들을 테스트한다. 또한 최종적인 유저 테스트에서의 시나리오를 위한 요소들로서 사용될 예정이다.

Table 4. Test case1 – 관리자 인증 시스템

구성요소	설명
테스트 이름 (Test Title)	관리자 인증
사전 조건 (Precondition)	1. 사전에 설정된 관리자 비밀번호가 있다. 2. 관리자들은 그 비밀번호를 알고 있다. 3. 특정 기능들에 관리자 인증 시스템이 적용된다.
수행 절차 (Test Step)	1. 관리자 기능에 접근 2. 비밀번호 입력 3. 비밀번호 일치 확인
기대되는 결과 (Expected Result)	비밀번호가 일치되면 관리자 전용 기능 인터페이스 오픈

Table 5. Test case2 – 관리자 기능 사용 예시

구성요소	설명
테스트 이름 (Test Title)	배경 전환
사전 조건 (Precondition)	관리자 인증 통과
수행 절차 (Test Step)	1. 관리자 기능 사용 2. 전환할 배경 선택 3. 확인

기대되는 결과 (Expected Result)	원하는 배경으로 전환
------------------------------	-------------

Table 6. Test case3 – 유저 기능 사용 예시

구성요소	설명
테스트 이름 (Test Title)	유저 아바타 세팅
사전 조건 (Precondition)	유저 전용 축제 아바타 제공
수행 절차 (Test Step)	유저용 아바타 선택
기대되는 결과 (Expected Result)	유저 전용 축제 아바타로 전환

12 Development Plan

12.1 전체 시스템 Integration

각각의 scene 과 세부기능을 통합하기 위해 게임 오브젝트와 컴포넌트, 그와 관련된 property 를 프리팹의 형태로 저장할 것이다. 각 프리팹을 main scene 에 배치하고, 조정하여 사용할 수 있도록 인스턴스화할 것이다.

GitHub 플랫폼을 사용하여 전체 시스템 통합 및 관리를 진행할 것이다. 이는 분산 버전 관리 툴인 깃(Git)을 사용하는 프로젝트를 지원하는 웹호스팅서비스로, 본 프로젝트의 협업을 돕는다.



Figure 10. GitHub

12.2 실시간 스트리밍 스크린

AVPro 는 Udon 에서 제공하는 비디오 플레이어 프리팹이다. 비디오를 업로드하고, 동시 접속 중인 사용자에게 비디오 싱크를 맞춰준다. YouTube Live, Twitch 및 기타 플랫폼과 같은 다양한 플랫폼에서 라이브 스트림을 지원한다.

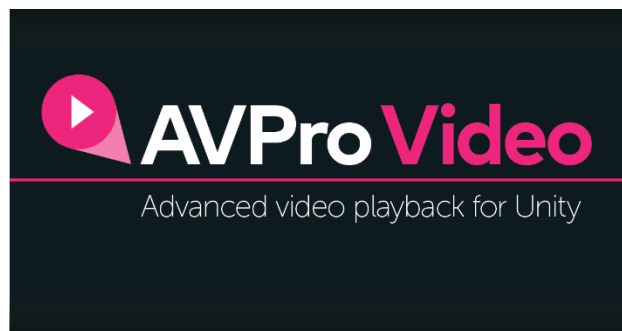


Figure 11. Unity Asset Store

12.3 축제 분위기 전환

사용자 인증 시스템을 Unity Prefab 형태로 저장하여 게시판 시스템과 결합할 예정이다.

하늘 배경 전환에 사용되는 skyboxes rendering material 에서는 unity asset store 에서 제공하는 무료 asset 을 사용할 것이다.



Figure 12. Unity Asset Store

파티클은 unity asset store 에서 제공하는 무료 asset 과 직접 custom 한 파티클 시스템을 모두 사용할 것이다.

배경음악은 저작권이 없는 무료 음원을 유튜브 오디오 라이브러리에서 다운받아 사용한다.



Figure 13. Youtube Audio Library

관리자가 축제 분위기 전환 버튼을 사용하였을 때, VRChat SDK3 의 udon behaviour 트리거 시스템을 사용할 것이다.

12.4 게시판

사용자 인증 시스템을 Unity Prefab 형태로 저장하여 게시판 시스템과 결합할 예정이다. 또한 Notice Board 에서 사용자와 상호작용하는 event post 의 경우, VRChat SDK 의 Udon Node Graph 를 이용하여 구현하고자 한다. Udon Node Graph 는 각 node 를 drag & drop 하는 식으로 편리하게 기능을 구현할 수 있다.



Figure 14. Udon Alpha

또한 C# script 를 udon 으로 변환할 수 있는 'Udon Sharp' 라이브러리를 사용하여 짧은 시간 내에 확장성 있는 기능을 구현할 수 있을 것이다. 기능이 복잡한 경우, Udon node graph 로 표현하면 오히려 가독성이 떨어져 불편할 수 있기 때문에 C# 스크립트를 작성하고 udon 으로 변환하는 과정이 필요할 수 있다.



Figure 15. Udon Sharp

또한 Guest Bulletin Board 를 구현할 때, 다수의 사용자가 메시지를 읽고 쓸 수 있어야 하기 때문에, Chatting server-client 가 구축되어 있는 asset 을 사용할 것이다. Photon chat 이나, 다른 사용자가 만든 무료 asset 들이 존재하기 때문에 그런 asset 을 기반으로 UI 를 변경하고 필요한 기능을 추가해 나가려고 한다.

12.5 이벤트 부스

아래와 같은 Udon Event Node 를 사용하여 구현할 것이다.

- OnPlayerJoined

유저가 instance 에 참여하면 발동한다. 참여한 유저가 출력된다.

- OnPlayerLeft

유저가 instance 를 나가면 발동한다. 나간 유저가 출력된다.

- OnStationEntered

유저가 물체의 스테이션(station)에 들어갈 때 발동한다.

- OnStationExited

유저가 물체의 스테이션(station)을 나을 때 발동한다.

12.6 축제용 아바타

사용자 인증 시스템과 입장 환영 메시지 시스템은 C# 코드를 Udon 어셈블리로 컴파일할 수 있는 UdonSharp 를 사용하여 구현된다. 아바타는 무료 아바타 asset 을 활용할 계획이며 시간적 여유가 있다면 아바타에 추가적으로 맞춤 제작한 악세서리를 장착하여 축제용 아바타의 특성을 부각하도록 한다.

13 Index

13.1 Tables

Table 1	38
Table 2	40
Table 3	42
Table 4	53
Table 5	53
Table 6	54

13.2 Figures

Figure 1	13
Figure 2	13
Figure 3	14
Figure 4	15
Figure 5	16
Figure 6	17
Figure 7	17
Figure 8	18

Figure 9	52
Figure 10	55
Figure11	55
Figure12	56
Figure13	56
Figure14	57
Figure15	57

13.3 Diagrams

Diagram 1	20
Diagram 2	21
Diagram 3	21
Diagram 4	22
Diagram 5	23
Diagram 6	24
Diagram 7	25
Diagram 8	26
Diagram 9	26

Diagram 10	27
Diagram 11	28
Diagram 12	29
Diagram 13	29
Diagram 14	30
Diagram 15	31
Diagram 16	32
Diagram 17	33
Diagram 18	33
Diagram 19	34
Diagram 20	35
Diagram 21	36
Diagram 22	37
Diagram 23	38
Diagram 24	39
Diagram 25	40
Diagram 26	41
Diagram 27	42
Diagram 28	43

Diagram 29	-----43
Diagram 30	-----44
Diagram 31	-----45
Diagram 32	-----45
Diagram 33	-----46
Diagram 34	-----47
Diagram 35	-----47
Diagram 36	-----48
Diagram 37	-----49