

협동 분산 시스템

Term Project Final Report



임민규 교수님

14조

컴퓨터공학부 201413231 류연수

컴퓨터공학부 201710551 오형석

컴퓨터공학부 201711420 임세빈

컴퓨터공학부 201113275 정준호

목차

Team member information and role	1
Project Goal	2
2.1. Precondition	2
2.2. Stack	2
2.3. 용어 정리	3
2.4. Top Level Use Case Diagram	3
Project Design	4
3.1. Project Structure	4
3.2. Server	4
3.2.1. CDS_Server	4
3.2.2. CDS_ServerEventHandler	4
3.2.3. CMServerSessionStub	5
3.3. Client	6
3.3.1 LoginActivity	6
3.3.2 SignUpActivity	6
3.3.3 MainActivity	7
3.3.4 StreamerListAdapter	7
3.3.5 ViewerActivity	7
3.3.6 StreamerActivity	7
3.4 Streaming Service	8
Project Result	9
Video and Github URL	9
Reference	10

1. Team member information and role

이름	류연수	오형석	임세빈	정준호
주역할	CM 이벤트 API	모바일 영상 송수신	모바일 GUI 개발	
부역할	문서 및 동영상 편집	모바일 채팅	문서 및 동영상 편집	
	팀장	로그인/회원가입	시연 결과 정리	
책임	33.3%	33.4%	33.3%	0%

2. Project Goal

본 프로그램은 하나의 클라이언트가 다른 클라이언트에게 실시간 스트리밍 영상을 제공하는 동영상 송수신 서비스입니다. 동영상 스트리밍과 실시간 멀티 채팅 기능을 제공합니다. CM이라는 미들웨어와 Web RTC 기반 미디어 서버를 활용하여 실시간 영상 스트리밍 어플리케이션을 구현하는 것이 목표입니다. 모든 클라이언트는 안드로이드로 구현되고, 서버는 Java와 CM, 미디어 서버는 Kotlin으로 구축됩니다.

2.1. Precondition

- 1) 모든 클라이언트는 단 하나의 세션에 소속되어 있다.
- 2) 모든 클라이언트는 스트리머와 시청자 중 어떤 역할도 할 수 있지만 한 번에 한 가지의 역할만 할 수 있다.
- 3) 한 세션 안에서 스트리머는 스트리밍을 시작한 클라이언트가 되며 스트리밍 중간에 스트리머가 바뀔 수 없다.
- 4) 스트리머와 시청자의 역할은 방으로 구성된 한 세션 내에서만 구분되며 방을 벗어난 이후에는 모두 동등한 클라이언트의 입장이다.
- 5) 동시에 여러명이 스트리밍을 할 수 없다.
- 6) 본 프로그램은 안드로이드에서만 구동 가능하며 다른 환경에서의 성능은 보장되지 않는다.

2.2. Stack

Server: Java, CM Project, Kotlin, WebRTC

DB: MySQL

Client: Android

2.3. 용어 정리

명칭	내용
시청자	동영상을 수신 받는 클라이언트
스트리머	동영상을 송신하는 클라이언트
스트리밍	스트리머가 해당 세션의 사람들에게 영상을 송신하는 행위
방	스트리머가 연 새로운 세션으로 하나의 스트리머와 다수의 시청자로 구성됨
채널	데이터를 주고받는 통신 망
세션	같은 방에 있는 클라이언트들을 묶어주는 통신 단위

2.4. Top Level Use Case Diagram

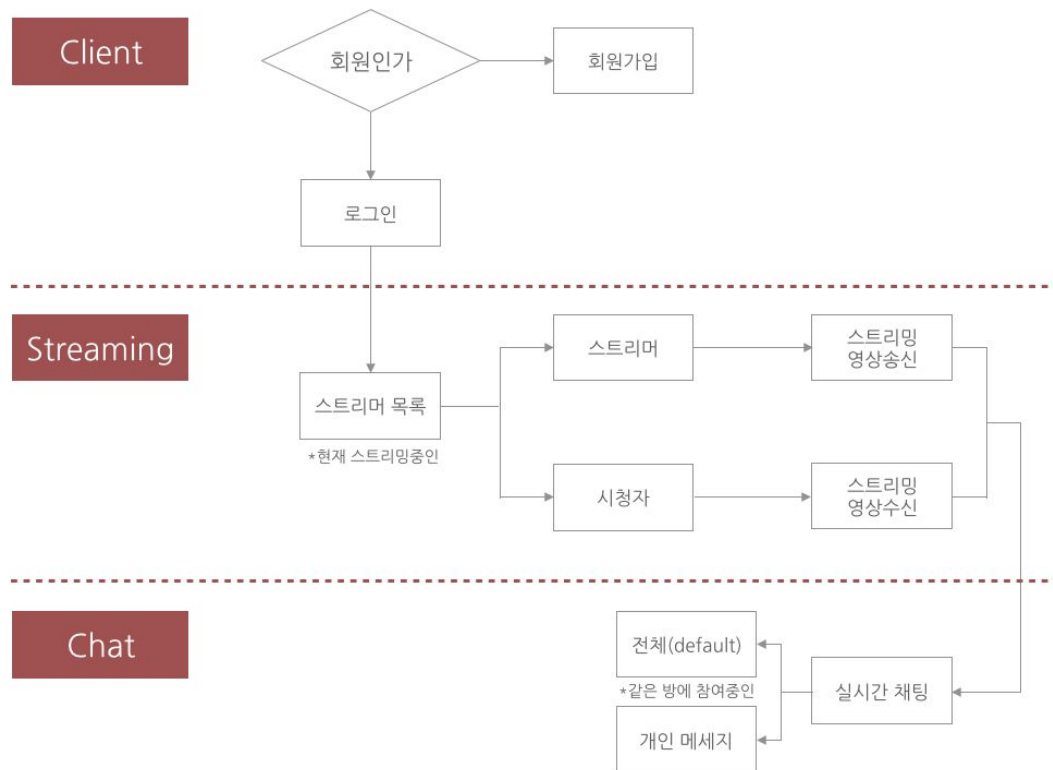


figure 1. Top Level Use Case Diagram

3. Project Design

3.1. Project Structure

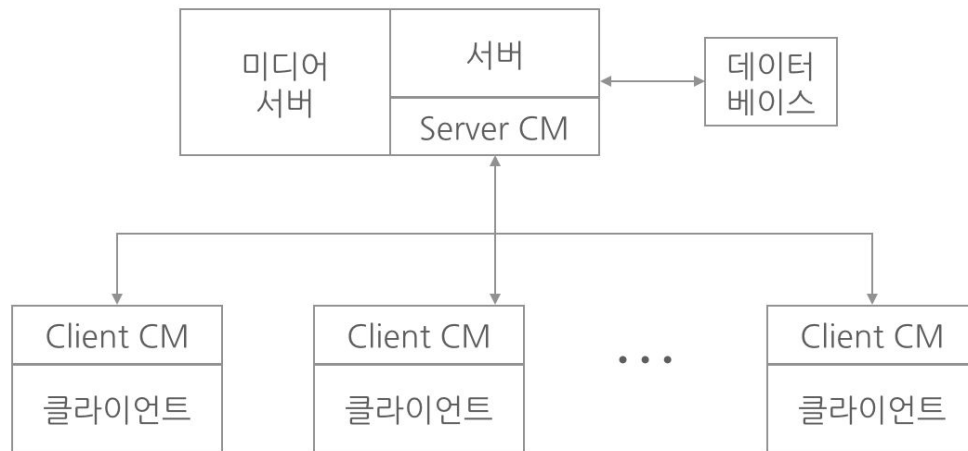


figure 2. Project Structure

3.2. Server

서버는 CM기반 서버와 미디어 서버로 구성됩니다. 모든 통신에 관여하며 클라이언트를 관리합니다. 미디어 서버는 스트리밍에서 수신 받는 동영상을 해당 세션의 시청자에게 송출하고, CM기반 서버는 각 세션 내의 클라이언트끼리 공유되는 채팅 서비스를 제공합니다. 서버는 하나의 스트리밍 세션만 제공하며, 스트리밍 세션은 디폴트 채널로 통신합니다.

3.2.1. CDS_Server

CM을 이용하여 구축된 서버 클래스입니다.

3.2.2. CDS_ServerEventHandler

서버로 오는 이벤트를 관리합니다.

	필요한 CM API	활용
세션 관리 (SessionEvent)	LOGIN	CMDBManager.authenticateUser을 통해 권한 확인 후 로그인 성공 및 실패 여부를 전송합니다.
	REGISTER_USER	회원이입시 DB에 유저 정보를 등록합니다.

	REQUEST_SESSION_INFO	모든 세션에 대해 해당하는 스트리머 ID를 SessionStub.getStreamerID()의 반환값으로 더미 메시지를 통해 전송합니다.
	JOIN_SESSION	유저를 세션에 등록합니다.
	LEAVE_SESSION	유저를 참여중인 세션에서 제외시킵니다.
채팅 서비스 제공 (InterestEvent)	USER_TALK	요청받은 채팅메세지를 해당 세션에 참여중인 모두에게 전송합니다.
그밖의 통신 (DummyEvent)	STREAMINGSTART	SessionStub.getPossibleSession()을 통해 이용 가능한 세션 이름을 유저에게 전송하고 모든 클라이언트에게 스트리머 목록 정보를 업데이트 하라는 메시지를 전송합니다.
	STREAMINGEND	SessionStub.leaveSession()을 이용하여 해당 세션의 정보를 업데이트 하고 모든 클라이언트에게 스트리머 목록 정보를 업데이트 하라는 메시지를 전송합니다.

3.2.3. CMServerSessionStub

모든 세션에 대해 그와 연관된 스트리머 ID를 매핑하여 기록합니다. 자료형은 배열(ArrayList)을 이용하고 각 세션은 진행중인 스트리머가 없을 경우 “.”로 기록합니다.

메소드 명	매개 변수	반환값 및 기능 설명
getStreamerID	없음	모든 세션에 매핑되는 스트리머 ID를 @@로 구분하여 순서대로 기록. 하나의 스트링으로 제공.
setStreamerID	num: int, streamerID: String	num에 해당하는 세션 인덱스의 스트리머 ID를 streamerID로 지정.
leaveSession	streamerID: String	세션을 순서대로 순환하며 streamerID와 동일한 스트리머 ID를 가지는 세션의 스트리머 ID를 “.”로 전환 후 해당 세션 이름을 제공.
getPossibleSession	senderID: String	세션을 순서대로 순환하며 스트리머 ID가 “.”인 세션의 이름을 제공.

3.3. Client

Client는 스트리밍 목록을 확인할 수 있고 스트리밍과 시청 중 한 가지 모드를 선택할 수 있습니다. Client가 스트리밍 모드이면 동영상을 송출하고, 시청자 모드이면 동영상을 수신합니다. 동영상이 송수신되는 동시에 시청자들은 채팅을 보낼 수 있으며, 스트리머와 시청자들은 채팅을 읽을 수 있습니다.

- ◉ 아래부터 CM API에 대한 내용은 **붉은색 진한 글씨**로 표시합니다.

3.3.1 LoginActivity

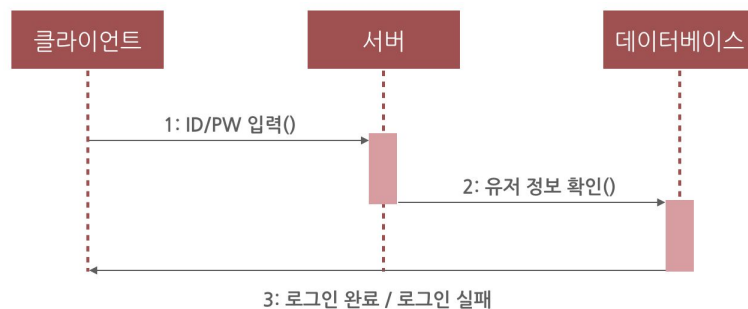


figure 3. 로그인 동작 시나리오

Client로부터 ID와 PW를 입력받아 서버에 전송합니다. 데이터베이스에 일치하는 정보가 있으면 로그인이 완료되고, 없으면 로그인에 실패합니다. 로그인 버튼이 눌렸을 때 리스너를 통해 이벤트를 감지하고 **syncLoginCM**을 실행시킵니다. 로그인 성공 여부를 확인 후 다음 페이지로 넘어갈 수 있도록 동기화된 CM API를 활용합니다.

3.3.2. SignUpActivity

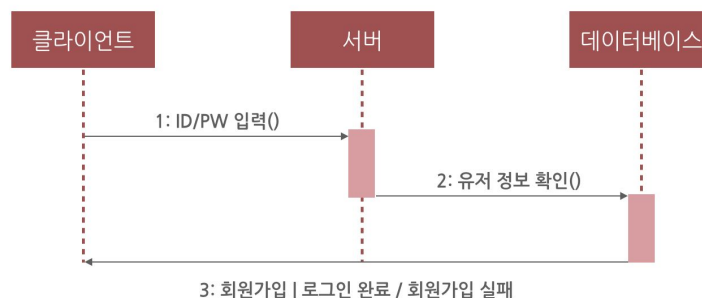


figure 4. 회원가입 동작 시나리오

회원가입 시에는 입력받은 ID가 데이터베이스에 존재하면 회원가입에 실패하고, 없으면 회원가입에 성공합니다. 추가로 회원가입시 비밀번호 대조를 통해 정확성을 확인합니다. 로그인과 마찬가지로 회원가입 버튼이 눌렸을 때 리스너를 통해 이벤트를 감지하고 **registerUser**를 통해 서버에 이벤트를 전송합니다.

3.3.3. MainActivity

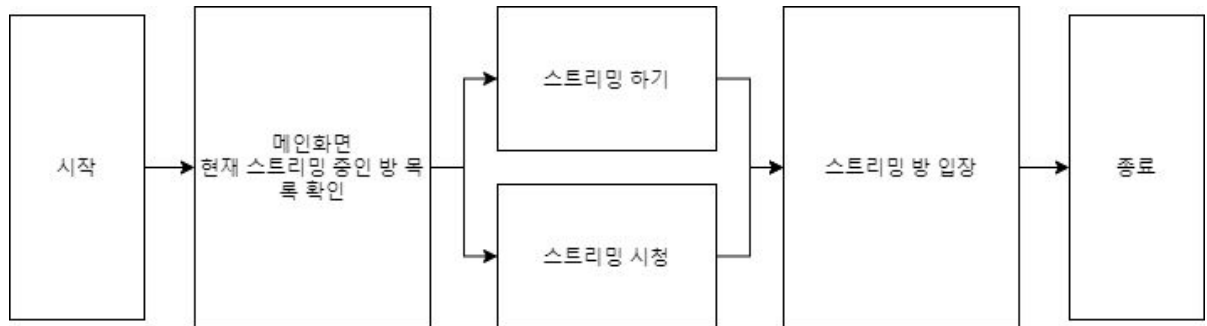


figure 5. 메인화면 동작 시나리오

Client가 프로그램을 실행하면 Main 화면을 볼 수 있습니다. Main 화면에는 어댑터를 통해 현재 스트리밍 중인 방의 목록을 확인할 수 있습니다. 스트리밍 목록 확인을 위해서 서버에 세션 정보를 요청해야 합니다. 이를 위해 메인 클래스에서 **requestSessionInfo**를 실행시킵니다. 스트리밍하기와 시청 중 선택할 수 있습니다. 스트리밍 버튼을 누르면 서버로 **STREAMINGSTART** 더미 이벤트를 전송하고 서버로부터 **RESPONSE_STREAMER_START**를 받아 **joinSession**를 호출하여 해당 세션의 스트리밍 화면으로 넘어갑니다. 로그아웃 버튼을 누르면 **logoutCM**이 호출되고 로그인 페이지로 넘어갑니다.

3.3.4 StreamerListAdapter

수직 그리드로 나뉘어 있는 방 중 하나를 선택하면 **joinSession**이 호출되며, 해당 방에 참여할 수 있습니다. 화면은 뷰어 페이지로 넘어갑니다.

3.3.5. ViewerActivity

방 참여자는 스트리머의 영상을 수신받고 채팅을 할 수 있습니다. **chat**을 통해 채팅 내용을 보내 해당 방 참여자들과 대화를 나눌 수 있습니다. 나가기 버튼이나 뒤로가기 버튼을 누르면 **leaveSession**을 통해 해당 세션에서 나가지며 화면은 메인 페이지로 넘어갑니다. 나가기 버튼 외에도 스트리머가 스트리밍을 종료하면 자동으로 해당 세션에서 나가지며 화면은 메인 페이지로 넘어갑니다.

3.3.6 StreamerActivity

스트리머는 개인 카메라를 통해 실시간 영상을 송신하고 방 참여자들의 채팅을 볼 수 있습니다. 나가기 버튼이나 뒤로가기 버튼을 누르면 **STREAMINGEND** 더미 메시지를 통해 해당 세션에 참여하는 모두에게 **RESPONSE_STREAMING_END**이 전송되며 해당 더미 메시지를 받은 클라이언트는 세션에서 나가지며 화면은 메인 페이지로 넘어갑니다.

3.4 Streaming Service

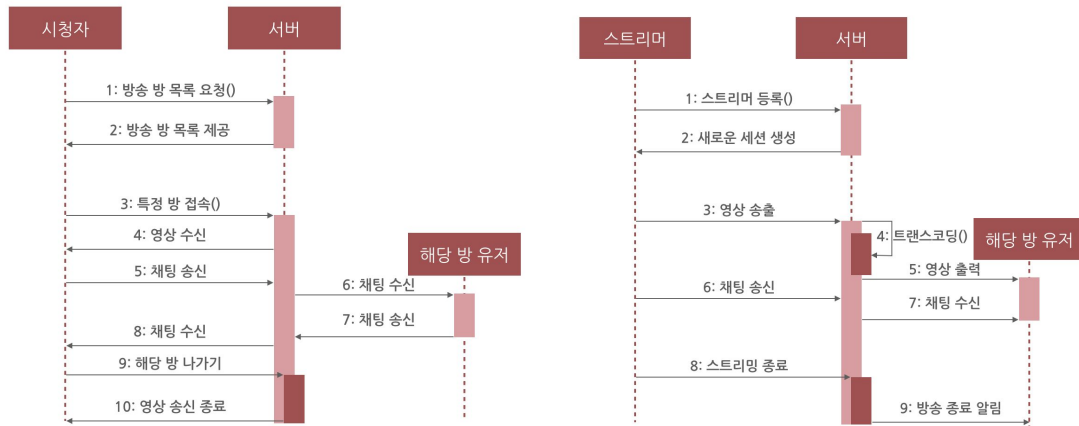


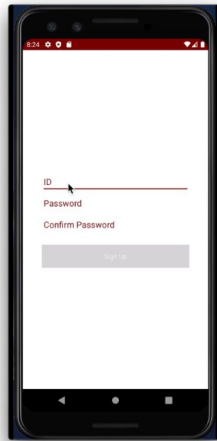
figure 6. 시청자/스트리머 동작 시나리오

Class	기능 설명
AppSdpObserver	SdpObserver를 상속 받은 객체
PeerConnectionObserver	PeerConnection의 Observer를 상속 받은 객체
RTCCClient	WebRTC 관리를 위한 객체
SignallingClient	Signaling Server와 통신을 위한 객체
SignallingClientListener	비동기적 통신을 위한 Listener 인터페이스

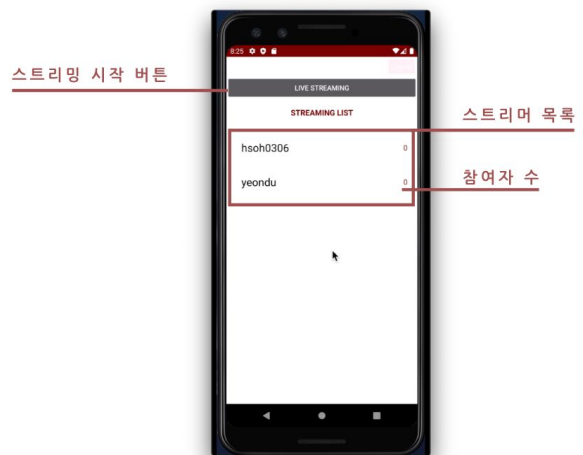
4. Project Result



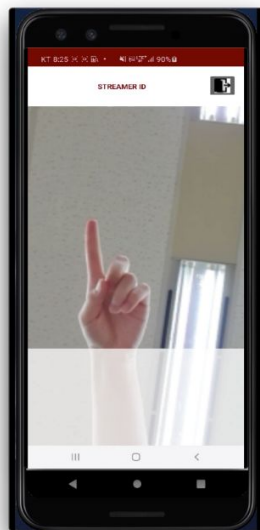
로그인 페이지



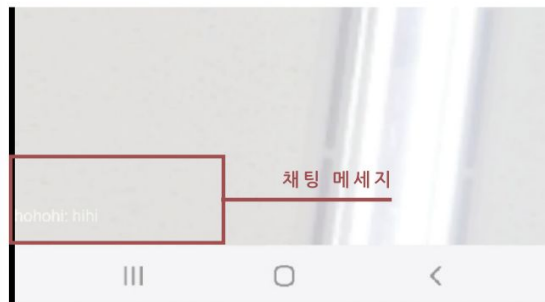
회원가입 페이지



메인 페이지



스트리머 시점



5. Video and Github URL

- 1) 결과 시연영상: <https://youtu.be/rm9pXmjovrA>
- 2) 결과 코드: https://github.com/hsoh0306/CDS_Project

Reference

- **Communication Framework (CM) User Guide**

- p.26** - Configuration of multiple sessions
- p.30** - CM event handler
- p.34** - Login to default server
- p.40** - Multiple sessions
- p.50** - A chatting event
- p.52** - Sending/receiving a CM event
- p.74** - Additional datagram(UDP) channels
- p.75** - Current client information
- p.85** - User management
- p.118** - CM DB management