|  |
| --- |
| Mr. Pres  상세 설계서 |
| 한성대학교 컴퓨터공학과 설계 프로젝트 |
| FrontFloor  엄두성 / 김강성 / 한종빈 / 조이슬 |

목차

[1 개요 4](#_Toc387533953)

[1.1 목적 4](#_Toc387533954)

[1.2 전체 시스템 구조 6](#_Toc387533955)

[2 구조 7](#_Toc387533956)

[2.1 용어 정리 7](#_Toc387533957)

[2.2 기능 요구사항 8](#_Toc387533958)

[2.3 전체 구조 8](#_Toc387533959)

[2.3.1 시나리오 8](#_Toc387533960)

[2.3.2 클라이언트 구조 9](#_Toc387533961)

[2.3.3 서버 구조 13](#_Toc387533962)

[2.4 기능 15](#_Toc387533963)

[2.4.1 로그인 15](#_Toc387533964)

[2.4.2 발표자료 업로드 16](#_Toc387533965)

[2.4.3 발표자료 실행 17](#_Toc387533966)

[2.4.4 리모트 컨트롤러 18](#_Toc387533967)

[2.4.5 화면 공유 19](#_Toc387533968)

[2.5 기능 요소별 모듈 테이블 20](#_Toc387533969)

[2.5.1 Audience 기능 모듈 테이블 20](#_Toc387533970)

[2.5.2 Controller 기능 모듈 테이블 21](#_Toc387533971)

[2.5.3 Speaker 기능 모듈 테이블 22](#_Toc387533972)

[2.5.4 COMMON 기능 모듈 테이블 23](#_Toc387533973)

[2.5.5 pptManage 기능 모듈 테이블 24](#_Toc387533974)

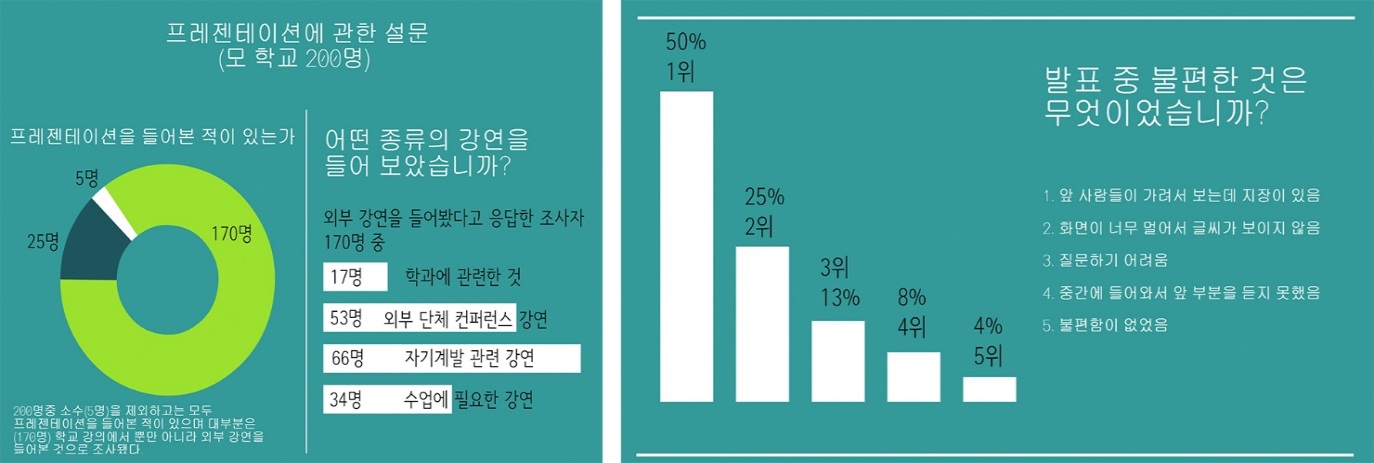
[2.5.6 view 기능 모듈 테이블 25](#_Toc387533975)

# 개요

Mr. pres는 PPT, PDF 파일을 웹에 업로드하여 사용자와 청중의 커뮤니케이션을 돕고 휴대용 디바이스를 통해 자체 슬라이드 전환 기능과 레이저포인터, 글씨 쓰기 기능을 지원해 기존의 교육용/발표용 터치스크린과 레이저포인터 장비의 비용을 줄이고 발표 자료의 공유와 청중과 발표자의 교류를 도모한다.

## 목적

최근 강연 시장의 급격한 성장과 다양한 경험 가능한 오픈형 국제 컨퍼런스, 강연식 방송 등의 등장으로 많은 사람들이 발표를 경험할 수 있는 길이 열렸다. 초, 중, 고등학생들의 프레젠테이션 경험 뿐만 아니라 대학생 단위의 전문적인 연합동아리의 성장으로 인해 학생단위로도 많은 컨퍼런스/발표를 들을 수 있게 되었다. **발표가 필요한 시장의 성장으로 인해 다수의 사람들을 상대로 강연을 하는 경우가 빈번해졌으며 또한 강연 시장에 대한 수요가 그대로 다수의 청중이 되며 다수의 청중에게 발표자가 효과적으로 의견을 전달하고 다양한 피드백을 받아 발표 자료의 질적 향상과 다수를 위한 발표 방식이 필요해졌다.**



<프레젠테이션에 관한 설문, 2014년 4월, SNS를 통해 학생 200명을 상대로 설문>

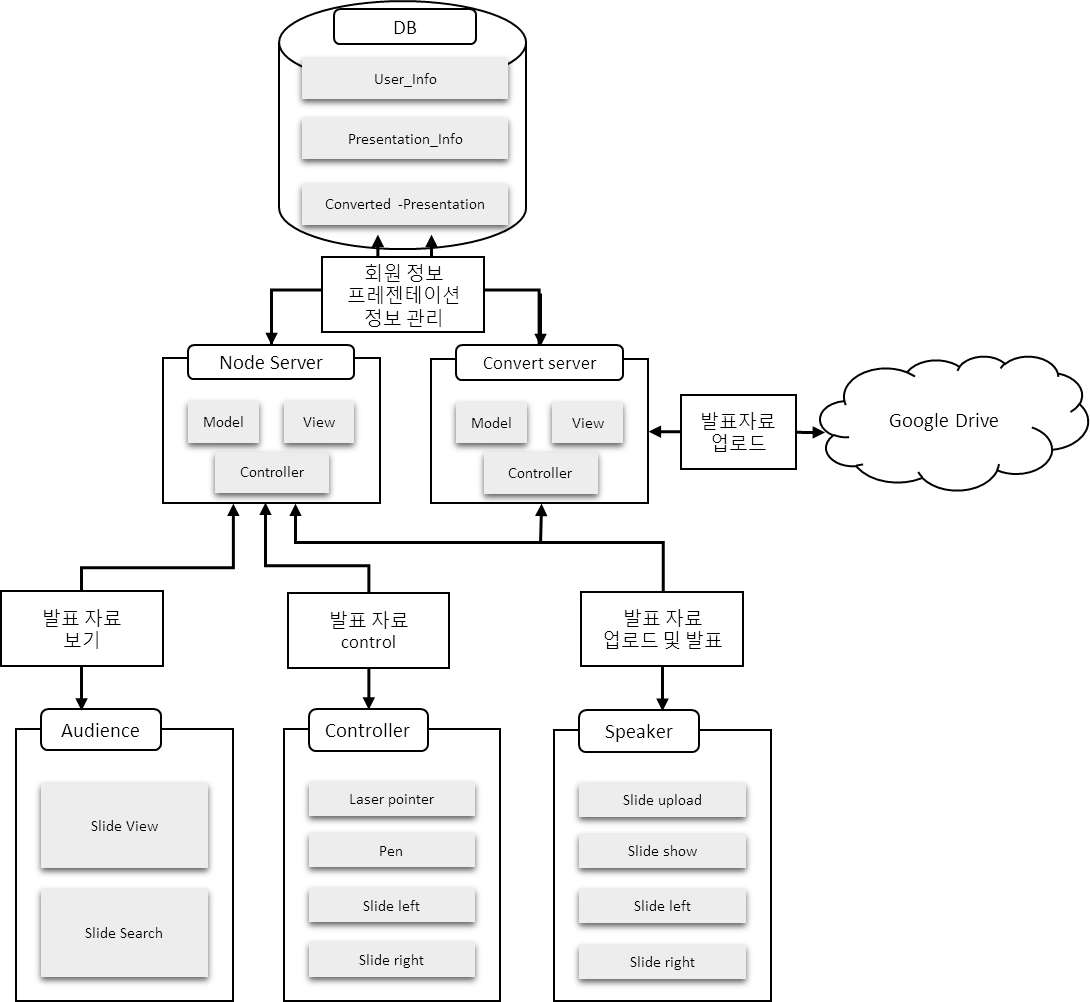
실제로 학생들을 상대로 조사해본 결과 200명 중 학교 수업 이외의 발표나 강연을 접한 학생들은 170명 (85%)이나 되었으며 그 중 119명은 학교와 관련 없는 외부 단체 컨퍼런스나 자기 개발에 관련한 강연을 접했다고 응답했다. 또한 수업에 필요한 강연이라고 답한 학생들 중 상당수는 연합동아리나 학과동아리 같은 동아리 활동에서 발표를 경험했다고 응답했다. 그 중 상당수는 같은 높이의 의자에 앉아 발표를 듣다 보니 앞 사람이 가려 발표 자료를 보기 어려웠다고 응답했고 그 다음으로는 화면이 너무 멀어서 발표자료를 보는데 어려움을 겪었다고 응답했다. 이는 현재 발표/강연시장, 학생 및 성인들의 모임에서의 발표가 양적으로는 성장하고 있지만 질적으로는 성장하지 못하는 것을 반증한다. 뿐만 아니라 학교의 수업에서도 같은 높이의 의자로 인해 발표 자료를 보기 어려워하는 경우가 있으며 시력이 안 좋거나 발표 장소가 넓을 경우 청자는 발표 자료를 접하는데 굉장한 어려움을 겪게 된다.

  
<실제 대학교의 수업 모습, 수업자료를 보기 어려울 뿐만 아니라 매우 멀리 느껴진다>

발표와 토론식 교육 중요성, SNS의 발달로 다양한 사람들이 모여 그룹을 이뤄 의견을 공유하는 모임들의 성장, 강연 시장의 발달로 인해 다양한 사람들이 발표의 수혜를 입게 되면서 이런 질적인 향상에 대한 수요가 급증하고 있다. 이것은 발표 자료가 개개인에게 모두 잘 보이는 동시에 발표 프로그램 사용자에게 정확하고 다양한 피드백을 통해 발표의 질적 향상을 해야 한다는 것이다. 현재 이것을 해결하기 위해서는 가장 대표적인 방법으로 각자 가지고 있는 휴대용 디바이스를 통해 발표 자료에 접근하여 발표를 감상/시청하는 것이며 보는 것뿐만 아니라 패블릿과 태블릿을 통해 발표프로그램 사용자에게 즉각적이고 다양한 피드백도 가능하게 할 수 있다는 의미이다.

## 전체 시스템 구조

위의 설계 목표를 만족시키기 위하여 Mr. Pres의 시스템 구조를 <그림1-1>과 같이 제안한다.



<그림 1-1> 전체 시스템 구조도

# 구조

이 장에서는 앞 장에서 설명한 설계 목표를 달성하기 위한 기능 요구 사항을 간략히 기술하고 그 기능을 구현하기 위한 개념적 구조를 기술한다.

## 용어 정리

|  |  |
| --- | --- |
| **용어** | **정의** |
| 사용자 | 회원과 비회원 모두를 이르는 말  회원 또는 비회원만 필요한 특정 경우는 따로 지칭하여 쓴다 |
| 회원 | 회원 가입 후 서비스를 이용하는 사람 |
| 비회원 | 방문자와 동일  회원 가입을 하지 않고 서비스를 이용하는 사람 |
| 프레젠테이션 | PPT 혹은 PDF, ODF 등의 형태로 제작되는 발표 자료 |
| 슬라이드 | 프레젠테이션의 각 페이지 |
| 슬라이드 쇼 | 전체 화면으로 프레젠테이션을 보는 것 |
| 발표 | 프레젠테이션의 싱크를 공유하거나 리모트 컨트롤러를 연결하는 것 |
| 발표자 | 특정 프레젠테이션을 발표할 수 있는 사람 |
| 싱크 공유 | 발표자 중 현재 프레젠테이션의 상태(보고 있는 슬라이드)를  다른 사용자에게 공유  싱크 공유자가 아닌 사용자는     싱크 공유된 프레젠테이션의 상태를 보거나     자신이 보고 싶은 페이지를 볼 수 있음 |
| 뷰어 | 사용자의 프레젠테이션을 볼 수 있는 기능을 지칭 |

## 기능 요구사항

웹 브라우저 기반 소프트웨어로서 Mr. Pres는 다음 표와 같은 총괄적인 기능들을 요구하고 있다. 이 총괄 기능 정의는 다음 절의 Mr. Pres의 개념적 총괄 구조를 설명하기 위한 것이며 각 총괄 기능에 대한 세부적인 기능은 제3장 Mr. pres의 기능에서 제시된다. 제2장의 나머지 부분에서는 각 기능을 구현하기 위한 개념적 세부 구조를 기술한다

|  |  |
| --- | --- |
| **기능** | **기능의 간략한 설명** |
| SNS 로그인 | 구글 API를 통한 로그인 기능 |
| 발표 자료 업로드 | PPT, PDF, ODP 형식의 발표자료를 업로드 할 수 있는 기능. |
| 발표 자료 저장 | 업로드할 자료를 사용자의 구글 드라이브에 저장해주는 기능 |
| 발표 자료 실행 | 업로드 된 자료를 보여주는 기능 |
| 리모트 컨트롤러 | Mobile Device를 통하여 PC의 발표자료를 컨트롤 하는 기능 |
| 화면 공유 | 발표자의 현재 발표 슬라이드를 자신의 PC에서 실시간으로 볼 수 있는 기능 |

<표 2-1> Mr. Pres의 기능

## 전체 구조

### 시나리오

1. 사용자:

Mr. Pres서비스를 통해 회원 가입을 할 수 있고, SNS 계정을 통한 회원 가입 또한 할 수 있다.

1. 회원:
   1. 발표 자료 업로드, 리모트 컨트롤러와 화면 공유 사용 발표

A씨는 어느 컨퍼런스에서 발표를 할 예정이다.  
발표를 위해 Mr. Pres서비스에 미리 발표 자료를 업로드해 두었다.  
발표 당일, A씨는 Mr. Pres서비스에 접속하여 올려 놓은 발표 자료를 확인하였다.  
발표 시간, 발표 장소는 매우 넓어 발표를 시작할 때 발표자 옵션에서 화면 공유 기능과 리모트 컨트롤러 기능을 활성화 하고, 발표를 시작하였다.  
발표를 시작하면서 스마트폰을 꺼내어 리모트 컨트롤러를 실행하여 발표자료와 연결하고, 발표를 듣는 청중들에게 Mr. Pres에 접속하여 슬라이드를 볼 수 있는 방법을 안내하였다.

* 1. 기존 업로드 발표 자료 사용 발표

B씨는 어느 소모임에서 팀원 A씨가 발표한 프레젠테이션을 발표하게 되었다.  
B씨는 Mr. Pres서비스에 접속하여 A씨 이름으로 발표 자료를 검색하였다.  
검색 된 자료 결과 중 A씨의 해당 자료를 선택 후 프레젠테이션 시작 버튼을 눌러 발표를 시작 하였다.

1. 비회원(회원도 가능):
   1. 발표자와 화면 공유가 되는 발표 듣기

C씨는 어느 컨퍼런스에 A씨의 발표를 들으러 갔다. 서둘러 갔지만 예상보다 늦게 도착한 C씨는 상당히 뒤에 앉는 바람에 A씨의 발표 자료가 잘 보이지 않아 발표를 듣지 않을까 했지만, A씨가 Mr. Pres서비스에 접속하여 발표 자료를 검색하여 접속하면 현재 발표자가 발표중인 화면을 실시간으로 볼 수 있다고 안내해주어 발표에 집중할 수 있었다.

* 1. 발표 자료 보기

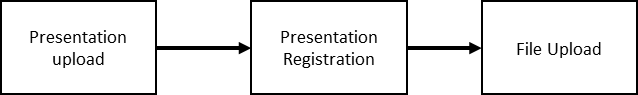
지난주에 발표한 자료가 다시 보고 싶어진 C씨는 Mr. Pres서비스에 접속하여 발표 자료를 검색하여 해당 자료를 다시 볼 수 있었다.

### 클라이언트 구조

1. 발표자
2. 발표 자료 업로드
3. 설계 목표

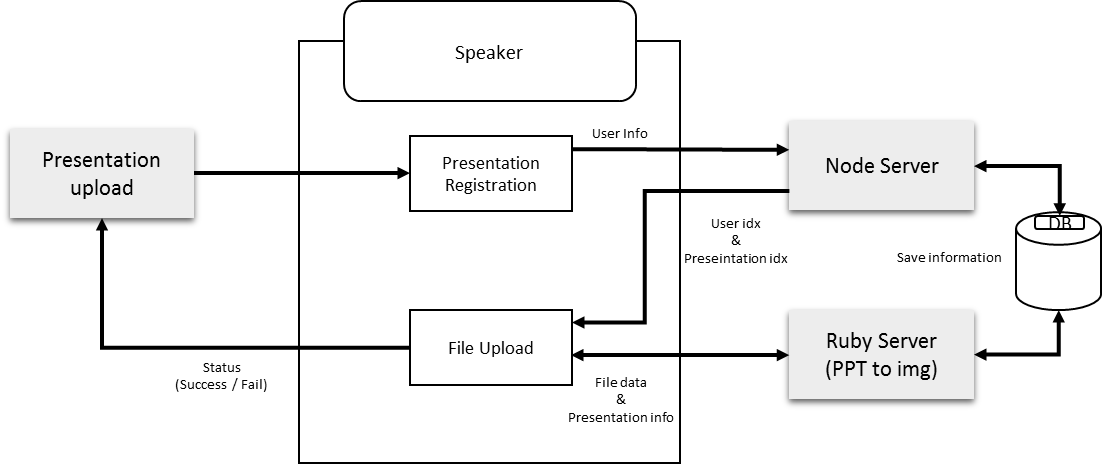
발표자료는 프레젠테이션 발표를 하기 위한 것이다. 발표자가 올린 파일과 발표자의 정보를 가지고 파일을 업로드하도록 구현한다.

1. 구조



발표자료 업로드에서 모든 과정은 프레젠테이션 등록(Node server) 후 해당파일을 업로드하는 방식으로 진행된다.

아래 그림에서 보면 Node서버의 결과(U-idx, P-idx)를 가지고 Ruby서버에 프레젠테이션 파일을 Mr.에 맞는 양식으로 ppt를 변환하는 요청을 처리한다. 처리결과는 파일이 정상적으로 업로드가 되었는지 실패하였는지 확인하는 status를 돌려주게 된다.

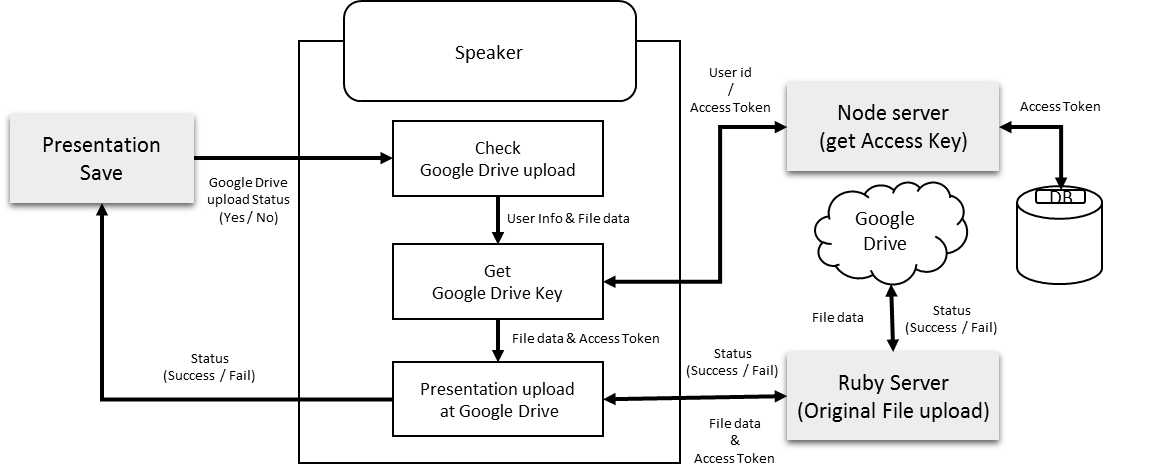


1. 발표 자료 저장
2. 설계 목표

발표자료를 발표자의 Google Drive에 upload하는 process는 사용자가 발표자료를 저장을 원하는 경우 파일저장을 제공하기 위한 것이다.

1. 구조

아래 그림에서 보면 우선 google drive에 프레젠테이션 파일을 업로드 할지를 확인 후 Google Key(Access Token)를 Node server에서 받아온다. 받아온 Access Token을 가지고 Ruby서버에서 Google Drive에 파일을 업로드를 요청 한다. Access Token 값이 없거나 Drive에 업로드를 실패 시 status는 fail을 return되고 성공 시 Success 를 돌려주게 된다.

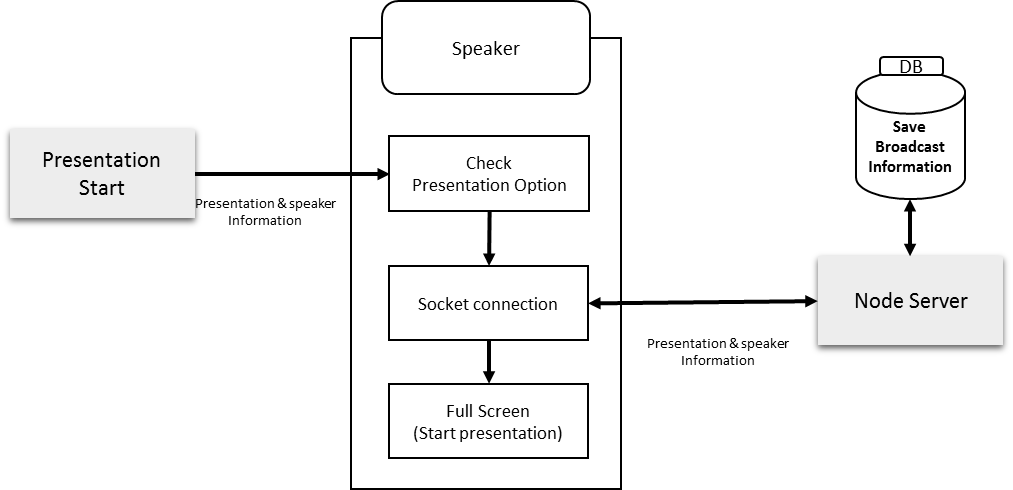


1. 발표 자료 실행
2. 설계 목표

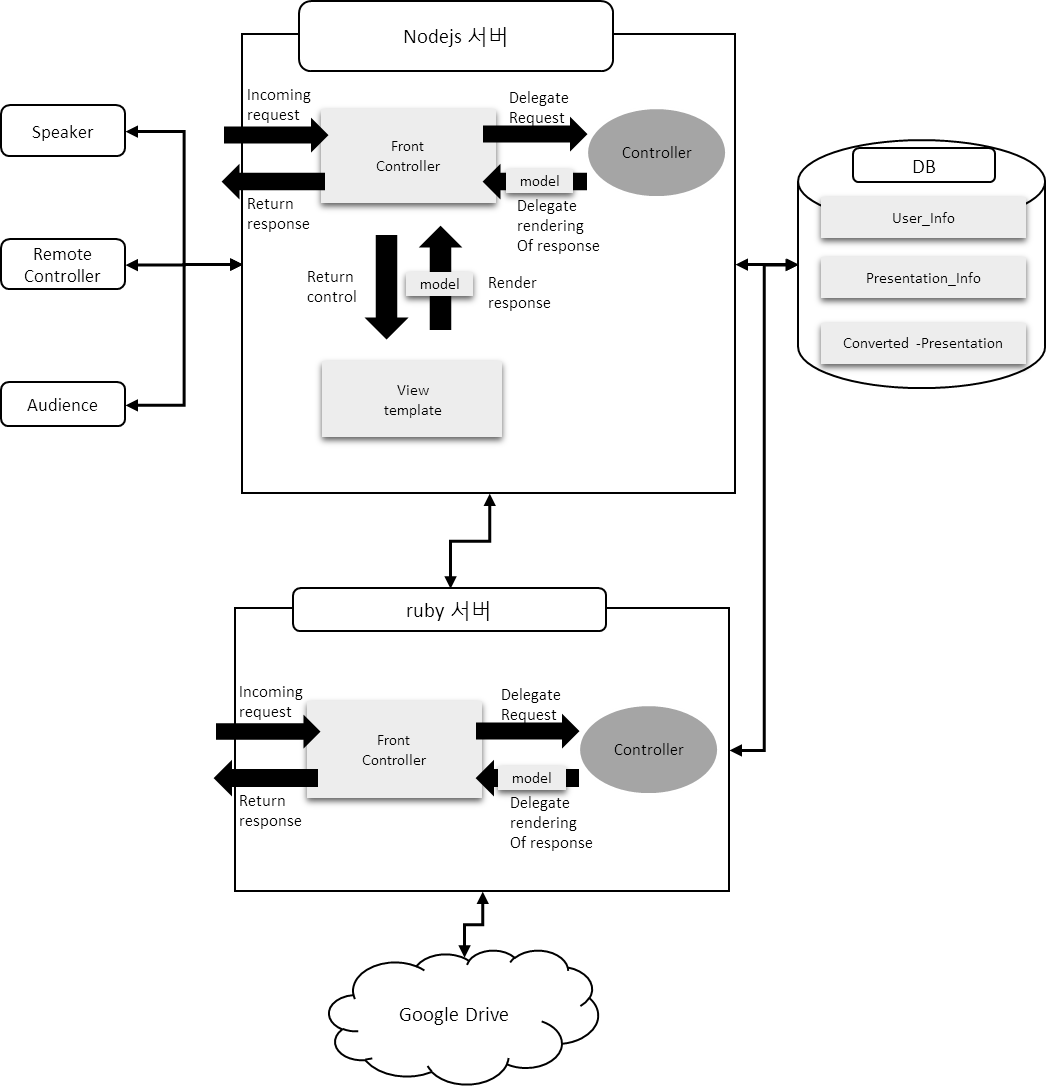
발표자의 Presentation을 제공하기 위함이다.

1. 구조

발표 시작을 하면 우선 발표 옵션을 확인한다. Option은 Sync 공유와 Remote Controller 사용 두 가지가 있다. 해당 옵션과 발표자료 정보를 node서버에 보내 socket을 연결을 한다. 하지만 option이 모두 꺼져있는 경우 socket연결하지 않는다. 그 후 발표를 할 수 있게 화면은 full screen으로 변경된다.



### 서버 구조

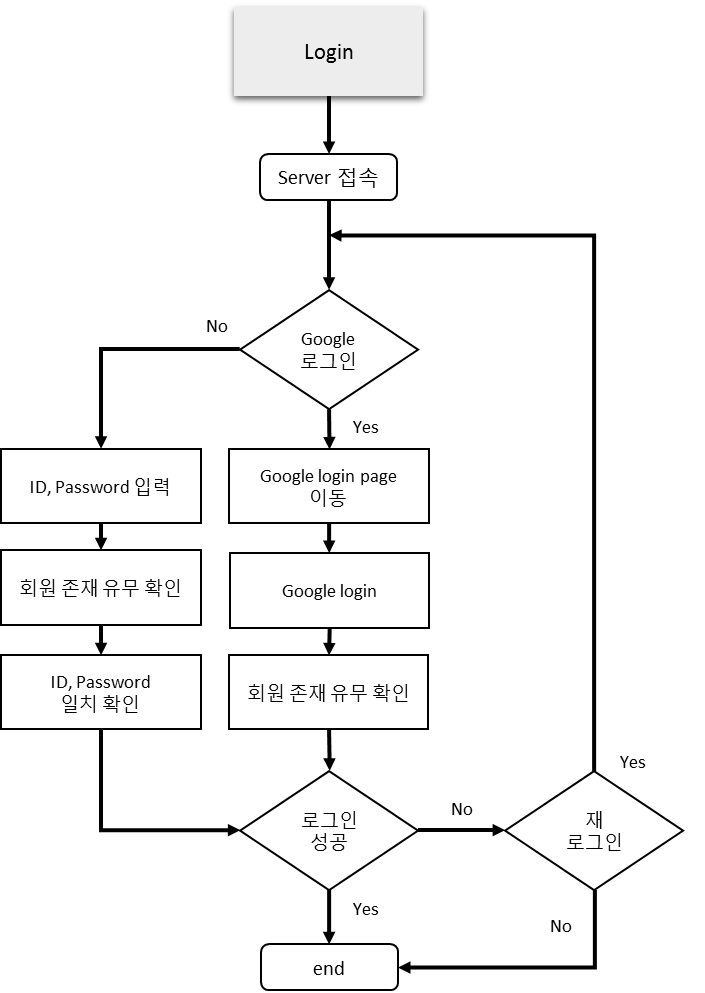


* + 1. Node 서버

1. Speaker가 화면공유 기능 사용 요청 시, 해당 기능 사용을 승인한다.   
   웹소켓을 사용하여 다른 사용자가 화면을 공유 받을 수 있도록 한다.
2. Speaker가 Remote Controller 사용 요청 시, 해당 기능 사용을 승인한다.  
   Speaker가 모바일 기기를 Remote Controller로 사용할 수 있도록 특정 페이지 주소를 리턴 해준다.  
   웹소켓을 사용하여 Speaker의 발표자료와 Remote Controller를 연결 시켜준다.
3. Remote Controller에서 슬라이드 이동 이벤트가 발생하면 Speaker의 발표자료에 해당 이벤트를 전달해준다.
4. Audience가 Speaker의 발표를 듣기 위해 발표자료에 연결을 요청하면 승인 한다.
5. DB 서버와 통신한다.
6. 사용자가 발표자료를 업로드하면 Ruby 서버로 해당 자료를 전달한다.
   * 1. Ruby 서버
7. node서버로부터 업로드 할 발표자료를 전달 받는다.   
   발표자료 업로드를 요청한 사용자의 구글 드라이브에 업로드한다.
8. 발표자료의 용량을 줄이기 위하여 발표자료를 이미지로 변환한다.
9. 변환 된 이미지를 서버에 저장한다.
   * 1. DB 서버
10. 회원가입 시 사용자의 정보를 저장한다.
11. 사용자가 업로드 한 발표자료 정보를 저장한다.
12. 변환 된 발표자료 이미지파일 정보를 저장한다.

## 기능

### 로그인



login\_1: 로그인을 google 로그인할지 local 로그인을 할지 선택한다

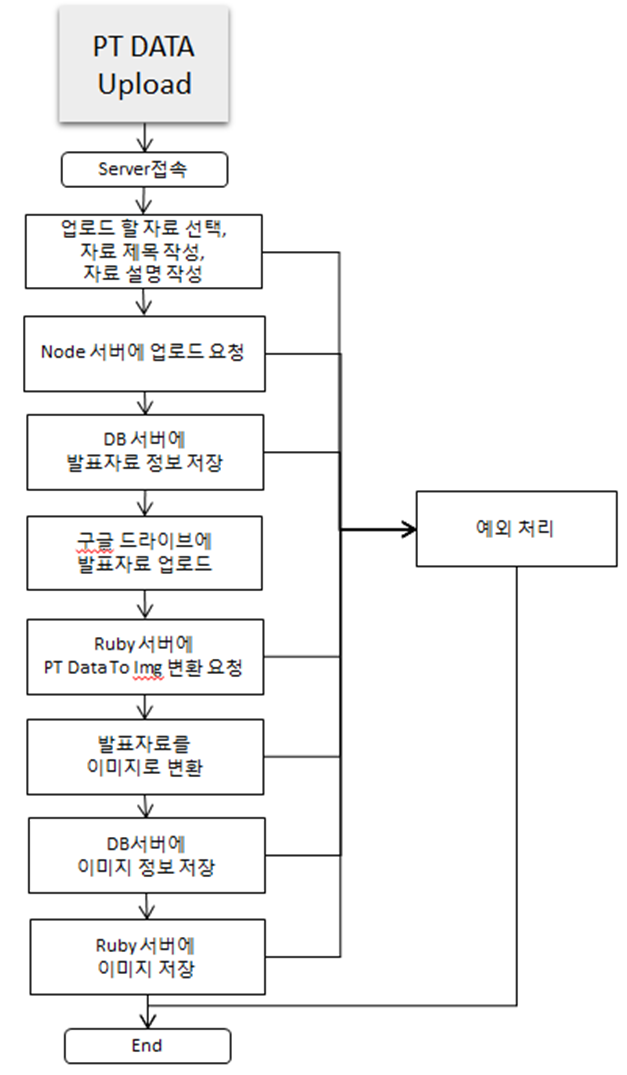
login\_2-1: google 로그인시 google페이지에 이동하여 google 로그인을 한다

login\_2-2: local 로그인시 자신의 ID Password를 입력한다.

login\_3-1: google 로그인에 해당 id가 Mr. Press의 사용자인지 확인한다. 존재 하지 않을 시 재 로그인 요청후 처리한다.

login\_3-2: local 로그인에 입력된 ID존재유무, Password 일치 여부를 확인한다. 존재하지 않는 id이거나 password가 틀릴 시 재 로그인 요청 후 처리한다.

### 발표자료 업로드



PTData\_Upload\_1: 업로드 할 파일을 선택한다. 업로드 할 자료의 제목과 설명을 작성 한다. 업로드 하는 파일의 파일형식이 지정된 형식과 다를 경우 에러 처리한다. 자료의 제목이 작성되지 않은 경우 에러 처리한다.

PTData\_Upload\_2: 선택된 파일과 제목, 설명을 Node 서버에 업로드 요청한다.

PTData\_Upload\_3: DB서버에 발표 자료 정보를 저장한다.

PTData\_Upload\_4: 업로드를 요청한 사용자의 구글 드라이브에 발표자료를 저장한다.

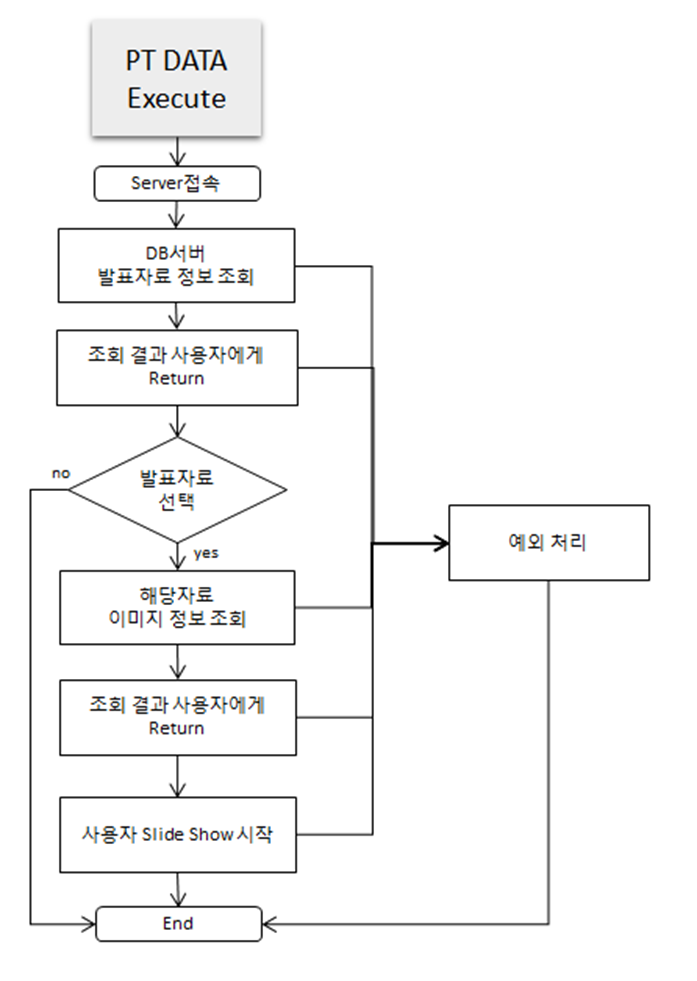
PTData\_Upload\_5: Ruby 서버에 발표자료를 이미지 파일로 변환 요청을 한다.

PTData\_Upload\_6: 발표자료를 이미지로 변환한다.

PTData\_Upload\_7: DB서버에 이미지 정보 저장

PTData\_Upload\_8: Ruby 서버에 이미지파일 저장.

### 발표자료 실행



PTData\_Execute\_1: DB서버에서 발표자료 정보를 조회한다.

PTData\_Execute\_2: 조회결과를 사용자에게 Return 해준다.

PTData\_Execute\_3: 사용자는 조회 결과 중 프레젠테이션에 실행하고자 하는 자료를 선택한다.

PTData\_Execute\_4: 사용자의 자료 선택 유무를 확인 한다.   
자료를 선택하지 않았다면 해당 기능은 종료 한다.

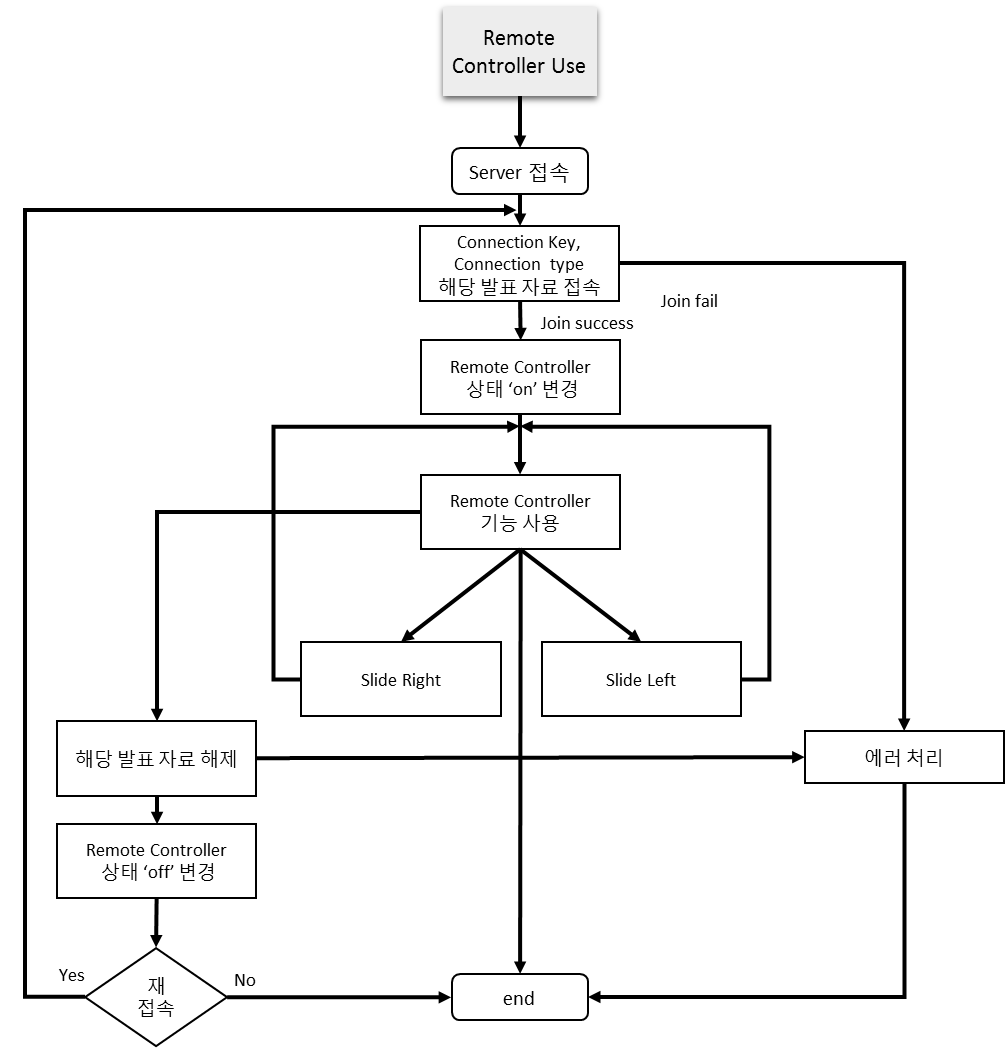
PTData\_Execute\_5: 사용자가 선택한 자료를 보여주기 위하여 DB서버에서 해당 자료의

이미지 정보를 조회 한다.

PTData\_Execute\_6: 이미지 파일을 Ruby 서버에서 사용자에게 Return 해준다.

PTData\_Execute\_7: 사용자는 Slide Show 누름으로써 해당 자료로 발표를 시작한다.

### 리모트 컨트롤러



REMOTECONTROLLER\_USE\_1: Connection key 값과 Connection Type을 가지고 해당 발표 자료에 접속한다. 접속 실패 시 에러를 처리한다.

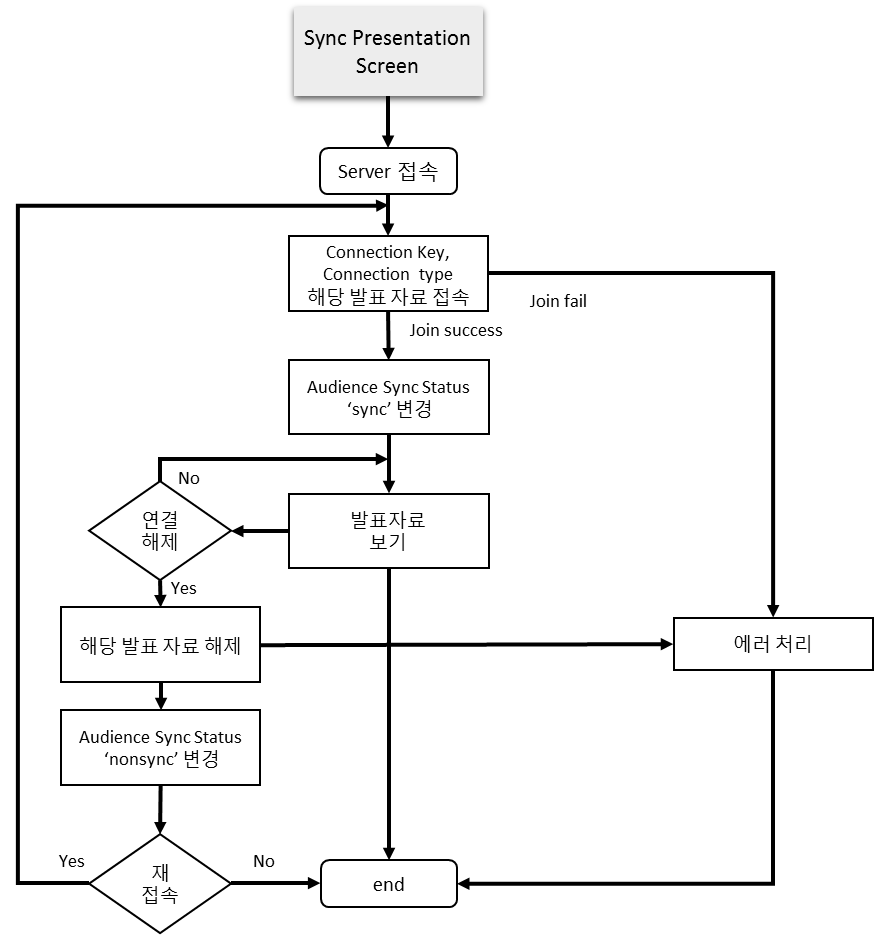
REMOTECONTROLLER\_USE\_2: Remote Controller의 상태를 ‘on’으로 변경한다.

REMOTECONTROLLER\_USE\_3-1: Remote Controller의 기능(slide left, right)을 반복하여 사용한다.

REMOTECONTROLLER\_USE\_3-2: Remote Controller의 연결을 해제 한다

REMOTECONTROLLER\_USE\_4: Remote Controller연결을 원할 시 처음(접속)부터 시작한다.

### 화면 공유



Sync\_Screen\_1: Connection key값과 Connection Type을 가지고 해당 발표자료에 접속한다. 접속 실패 시 에러를 처리한다.

Sync\_Screen \_2: Audience의 상태를 ‘sync’으로 변경한다

Sync\_Screen \_3-1: 발표자의 화면을 공유 받는다.

Sync\_Screen \_3-2: 발표자료의 연결 해제를 원할 시 발표자료의 접속을 해제한다.

Sync\_Screen \_4: 발표자료의 화면공유를 재 연결을 원할 시 처음(접속)부터 시작한다.

## 기능 요소별 모듈 테이블

### Audience 기능 모듈 테이블

|  |  |
| --- | --- |
| 모듈이름 | Audience |
| **서브모듈** | audience.init(), events.left(), events.right(), events.sync() |
| **목적** | Mr.Pres의 사용자 중 청중 관련 모듈 |
| **입력파일** | - |
| **출력파일** | - |
| **입력변수** | connectionKey, SyncObject |
| **출력변수** | - |
| **관련 모듈** | PtView, Speaker |
| **관련 기능요소** | view.move() |
| **기능 설명** | 1. 청자가 발표자의 발표자료에 연결을 요청하면 connectionKey의 유효성을 검사한 후 자료와 연결을 허용한다. 2. 연결 시 발표자가 현재 발표중인 슬라이드로 청자의 화면을 이동시킨다. 3. 동기화 상태 해제 키를 입력 할 시 자동 싱크 기능을 중단한다. 4. 동기화 상태를 활성화할 시 발표자가 현재 발표중인  슬라이드로 청자의 화면을 이동시킨다. 5. 발표자가 발표를 종료 할 시 청자의 화면에 발표가 종료 되었음을 알려준다. |

### Controller 기능 모듈 테이블

|  |  |
| --- | --- |
| **모듈이름** | **Controller** |
| **서브모듈** | joinRoom, leaveRoom, slideMove, penPointer |
| **목적** | Mr.Pres의 리모트 컨트롤러 관련 모듈 |
| **입력파일** | - |
| **출력파일** | - |
| **입력변수** | connectionKey, |
| **출력변수** | SyncObject |
| **관련 모듈** | PtView, Speaker, Audience |
| **관련기능요소** | view.move() |
| **기능 설명** | 1. 발표자의 connectionKey와 발표자료의 connectionKey를 확인하여 리모트 컨트롤러와 연결한다. 2. 리모트 컨트롤러에서 슬라이드 이동 이벤트가 발생 시, 해당 이벤트를 서버에게 전달하여 발표자료의 슬라이드가 이동하도록 한다. 3. 발표자의 현재 발표 진행 시간 표시한다. 4. 펜을 발표자료 화면에 표시한다. 5. 레이저 포인터를 발표자료 화면에 표시한다. |

### Speaker 기능 모듈 테이블

|  |  |
| --- | --- |
| **모듈이름** | **Speaker** |
| **서브모듈** | Speaker.init(), events.left(), events.right(), events.sync(), makeQRCode() |
| **목적** | Mr.Pres의 발표자 관련 모듈 |
| **입력파일** | - |
| **출력파일** | - |
| **입력변수** | userInfo |
| **출력변수** | connectionKey, SyncObject |
| **관련 모듈** | PtView, Audience |
| **관련기능요소** | view.move() |
| **기능 설명** | 1. 사용자가 발표자료를 선택 후 화면공유, 원격제어 기능을 활성화 시키면 사용자 정보를 이용하여 유일한 connectionKey를 생성. 2. 발표자가 슬라이드 이동 시, 발표자료가 이동한다. 3. 화면 공유 시, 슬라이드 이동 이벤트를 서버에게 전달하여 발표자료의 슬라이드가 이동하도록 한다. 4. 펜을 발표자료 화면에 표시한다. |

### COMMON 기능 모듈 테이블

|  |  |
| --- | --- |
| **모듈이름** | **COMMON** |
| **서브모듈** | common.init(), localLogin(), googleLogin(), join(), modify() |
| **목적** | Mr.Pres의 회원 관련 공통적으로 사용하는 부분을 관리하한다. |
| **입력파일** | - |
| **출력파일** | - |
| **입력변수** | email, password, name |
| **출력변수** | result - login | join 성공여부의 결과 |
| **관련 모듈** | - |
| **관련기능요소** | - |
| **기능 설명** | 1. 회원가입, 회원정보 수정 시 email 중복 및 password, 이름의 입력 값을 확인한다. 2. login의 종류는 google Login, local Login의 두 가지 종류가 있다. google로 로그인시 google 연결페이지가 나타나고, Local Login 시 DB에 접근하여 올바른 접속 정보인지 판단한다. |

### pptManage 기능 모듈 테이블

|  |  |
| --- | --- |
| **모듈이름** | **pptManage** |
| **서브모듈** | upload(), modify(), delete() |
| **목적** | 사용자의 발표자료를 업로드, 수정, 삭제 등 파일관리를 한다. |
| **입력파일** | ppt, pdf, odp등 발표자료 |
| **출력파일** | - |
| **입력변수** | title, description, userInfo |
| **출력변수** | result – 발표자료 업로드 | 삭제 | 수정의 성공여부 |
| **관련 모듈** | - |
| **관련기능요소** | - |
| **기능 설명** | 사용자가 발표자료의 title, description, 발표자료등을 기입하여 업로드시 파일을 서버에 전달한다. 만약 title을 미입력시 발표자료(파일)의 이름을 제목으로 대체하여 title로 지정한다.  사용자의 발표자료를 수정한다. 프레젠테이션의 파일자체는 바꿀 수 없으며, title 및 description의 내용만 바꿀수 있다.  사용자가 올린 발표자료를 삭제한다. 삭제 시 복구는 불가능하다. |

### view 기능 모듈 테이블

|  |  |
| --- | --- |
| **모듈이름** | **view** |
| **서브모듈** | move(), presentationFullScreen(), left(), right(), draw() |
| **목적** | 프레젠테이션 슬라이의 전반적인 이벤트를 담당하는 모듈이다. |
| **입력파일** | - |
| **출력파일** | - |
| **입력변수** | right, left |
| **출력변수** | result – 발표자료 업로드 | 삭제 | 수정의 성공여부 |
| **관련 모듈** | speaker, audience, controller |
| **관련기능요소** | - |
| **기능 설명** | 발표중인 상태에는 서버에서 온 값에 따라 left, right, draw등의 기능을 수행한다. 또한 발표 중이 아닌 상태에는 사용자의 입력 값에 따라 left, right의 입력을 수행한다.  left, right는 프레젠테이션의 1~페이지수 까지만 이동할 수 있다.  draw는 현재 슬라이드에 그림을 그리는 기능을 한다.  사용자가 전체화면을 누를 시 브라우저는 전체화면으로 변환된다. |