

자율주행기술개발혁신사업 경찰청 소관연구 2-1과제
자율주행 혼재 시 도로교통 통합관제시스템 및 운영기술 개발(2차)

자율주행 통합교통관제센터 표준검증플랫폼 설계서

KADIF-22-04-01-64-002

2022. 12

(사)한국지능형교통체계협회

<제목 차례>

1. 개요	1
1.1. 과업 개요	1
1.2. 설계 항목	1
2. 전용 컴파일러	2
2.1. 기능 설계	2
2.2. 출력 설정 기능	2
2.3. 블록 설계	3
2.3.1. Arguments Process Block	3
2.3.2. Parser Block	4
2.3.2.1. Lexical item	4
2.3.2.2. 기타	9
2.3.3. Syntax	9
2.3.3.1. X.680	9
2.3.3.2. X.681	23
2.3.3.3. X.682	24
2.3.3.4. X.683	24
2.3.4. Data Registry	25
2.3.5. Emitter Block	26
2.4. 기능 정의	27
2.4.1. Argument Process 기능	27
2.4.2. Data Registry 기능	27
2.4.3. Emitter 기능	27
2.5. 상세 설계	28
2.5.1. Information Base	28
3. DB 및 DBMS 시스템 구축	29
3.1. 사용자 기능	29
3.2. 관리자 일반 기능	30
3.3. 관리자 메타데이터 연동 기능	31
3.4. 관리자 시스템 연동 기능	31
3.5. 관리자 시스템 운영 통계 기능	31
3.6. S/W 설계	32
3.6.1. NginX(Web Server)	32
3.6.2. Gunicorn(WSGI Server)	32
3.6.3. Python Django(WSGI Application)	32
3.6.4. RDBMS(MySQL)	32
3.7. H/W(클라우드) 설계	33
3.7.1. VM(Virtual Machine) 기반 가상화 기술	33
3.7.2. Container 기반 가상화 기술	33
3.7.3. 가상화 기반 메타데이터 관리시스템 블록 구성도	33
3.8. DB 설계	34
3.8.1. 논리 모델(Logical Model) 설계	34
3.8.2. 물리 모델(Physical Model) 설계	35
3.8.3. 테이블 정의	36
3.8.3.1. User	36

3.8.3.2. UserDetail	36
3.8.3.3. Group	36
3.8.3.4. UserGroup	37
3.8.3.5. Permission	37
3.8.3.6. UserPermission	37
3.8.3.7. GroupPermission	37
3.8.3.8. ContentType	38
3.8.3.9. Category	38
3.8.3.10. Notice	38
3.8.3.11. NoticeDetail	38
3.8.3.12. Post	39
3.8.3.13. PostDetail	39
3.8.3.14. Comment	39
3.8.3.15. Metadata	40
3.8.3.16. MetadataDetail	40
3.8.3.17. MetadataRequest	40
3.8.3.18. MetadataLog	41
3.8.3.19. MetadataHistory	41
4. UI 설계	42
4.1. 사용자 페이지	42
4.1.1. 메인화면	42
4.1.2. 로그인 시 메인화면	43
4.1.3. 로그인	44
4.1.4. 회원가입	45
4.1.5. 공지사항	45
4.1.6. 로그인 시 공지사항	46
4.1.6.1. Meta data 조회	47
4.1.7. 로그인 시 메타데이터 수정/등록 영역	48
4.1.7.1. 메표준 자료실	49
4.1.8. 로그인 시 표준 자료실 화면	50
4.1.8.1. My Page	50
4.1.9. 관리자 페이지	53
4.1.9.1. 메인화면	53
4.1.9.2. 로그인	54
4.1.9.3. 사용자 관리	55
4.1.9.4. 그룹 관리	57
4.1.9.5. 메인화면 관리	59
4.1.9.6. 공지사항 관리	63
4.1.9.7. 표준 자료실 관리	64
4.1.9.8. 카테고리 관리	66
4.1.9.9. 메타데이터 관리	67
4.1.10. 사이트맵 설계	69
4.1.10.1. 사용자 페이지	69
4.1.10.2. 관리자 페이지의 사용자/그룹 관리	70
4.1.10.3. 관리자 페이지의 디스플레이 관리	70

5. SW 개발 환경	71
5.1. 메타데이터 관리시스템의 SW 개발 환경	71
5.2. SW 연동 방안	72
5.2.1. 전용컴파일러 연동 방안	72
5.3. 데이터 운영 방안	73
5.3.1. 데이터의 분류	73
5.3.1.1. 정형 데이터(Structured Data)	73
5.3.1.2. 비정형 데이터(Unstructured Data)	73
5.3.1.3. 반정형 데이터(Semi-Structured Data)	73
5.3.2. 메타데이터 관리시스템의 데이터 운영 방안	73
5.3.2.1. 정형 데이터(Structured Data) 운영 방안	73
5.3.2.2. 반정형 데이터(Semi-Structured Data) 운영 방안	73
6. 인터페이스	75
6.1. 기능 설계	75
6.1.1. Profiling 기능	75
6.1.2. Functionality Invocation 기능	75
6.2. 블록 설계	76
6.2.1. Interface Module Host	76
6.2.2. Interface Module Servant	76
6.3. 구현 설계	77
6.3.1. Host 초기화 기능	77
6.3.2. Servant 초기화 기능	77
6.3.3. Establishment & Negotiation	77
6.3.4. Polling	78
6.3.5. Functionality Invocation	78
6.4. 상세 설계	79
6.4.1. IPC Frame	79
6.4.2. Profile Syntax	79
7. 시나리오 엔진	80
7.1. 기능 설계	81
7.1.1. Protocol Spec Loading 기능	81
7.1.2. Upper Layer Interface 기능	81
7.1.3. Lower Layer Interface 기능	81
7.1.4. Control 기능	81
7.2. 블록 설계	82
7.2.1. Message Broker	82
7.2.2. Protocol Agent	82
7.2.3. CODEC Engine	83
7.3. 구현 설계	84
7.3.1. 초기화 기능	84
7.3.2. Message Broker	84
7.3.2.1. 초기화 기능	84
7.3.2.2. Message Routing 기능	84
7.3.2.3. External Asset 연동 기능	85
7.3.3. Protocol Agent	85

7.3.3.1. Capsulation 기능	85
7.3.3.2. Finite State Machine 기능	86
7.3.3.3. Control 기능	86
7.3.4. CODEC Engine	86
7.3.4.1. Encode 기능	86
7.3.4.2. Decode 기능	87
7.4. 상세 설계	88
7.4.1. Protocol Spec	88
7.4.1.1. Message Spec	88
7.4.1.2. Protocol Data Description	89
7.4.1.3. Protocol State Description	89
7.4.1.4. Timing Behavior	89
8. 시나리오 엔진 연동시스템	90
8.1. S/W Block 설계	90
8.1.1. S/W Block 구조	90
8.1.2. 모듈 별 기능 정의	90
8.1.2.1. tcp/ip 통신 모듈	90
8.1.2.2. 시나리오 엔진 접속 관리 모듈	90
8.1.2.3. Websocket 통신 모듈	91
8.1.2.4. 표준적합성 검증시스템 접속 관리 모듈	91
8.1.2.5. 프로토콜 변환 모듈	91
8.1.2.6. Event & Thread handling 모듈	91
8.1.2.7. Configuration 모듈	91
8.1.2.8. Common 공통 모듈	91
8.1.3. SW 개발 환경 및 사용 기술	91
8.1.3.1. SW 개발 환경	91
8.1.3.2. SW 사용 기술	91
8.1.4. SW 연동 방안	94
8.1.4.1. 표준적합성 검증시스템과 시나리오엔진 연동 절차	94
9. 표준적합성 검증시스템	95
9.1. S/W Block 설계	95
9.1.1. S/W Block 구조	95
9.1.2. 모듈 별 기능 정의	95
9.1.2.1. http 통신(사용자 요청 처리) 모듈	95
9.1.2.2. 인증 모듈	95
9.1.2.3. Websocket 통신 모듈	96
9.1.2.4. 시나리오엔진 연동시스템 접속 관리 모듈	96
9.1.2.5. 프로토콜 변환 모듈	96
9.1.2.6. Database ORM 모듈	96
9.1.2.7. Common 공통 모듈	96
9.1.3. SW 개발 환경 및 사용 기술	96
9.1.3.1. SW 개발 환경	96
9.1.3.2. SW 사용 기술	96
9.1.4. SW 연동 방안	100
9.1.4.1. 시나리오엔진 연동시스템 연동 절차	100

10. 사용자 매뉴얼	101
10.1. 표준검증플랫폼의 기능	101
10.2. 기능별 사용방법	102
10.2.1. 회원가입	102
10.2.2. 로그인	103
10.2.3. 마이 페이지	104
10.2.4. 비밀번호 변경	105
10.2.5. 회원탈퇴	106
10.2.6. 나의 글	107
10.2.7. 나의 메타데이터	108
10.2.8. 공지사항	109
10.2.9. 자료실	110
10.2.10. 게시판	111
10.2.11. 게시판 글 등록	112
10.2.12. 게시판 댓글	113
10.2.13. 메타데이터 목록 조회	114
10.2.14. 메타데이터 신청	115
10.2.15. 메타데이터 조회/수정/삭제	116
11. 운용자 매뉴얼	118
11.1. 목적	118
11.2. S/W 운영 환경	118
11.2.1. 메타데이터 관리시스템의 운영 환경	118
11.2.2. 메타데이터 관리시스템의 논리적 S/W 블록 구조	119
11.2.3. 메타데이터 관리시스템과 전용 컴파일러 연동 구조	120
11.3. S/W 설치	121
11.3.1. 메타필수 설치 조건(prerequisite)	121
11.3.2. Ubuntu 필수 패키지 설치	122
11.3.3. Conda 설치	122
11.3.4. MySQL 설치	122
11.3.5. NginX 설치	122
11.3.6. 메타데이터 관리시스템 설치	123
11.3.7. python 패키지 설치	123
11.4. S/W 설정 및 구동	124
11.4.1. 메타데이터 관리시스템 환경 설정	124
11.4.2. NginX 환경 설정 및 구동 스크립트	124
11.4.3. Gunicorn 환경 설정 및 구동 스크립트	125
11.4.4. 메타데이터 관리시스템의 관리자 계정 생성	127
11.4.5. 메타데이터 관리시스템의 구동 스크립트	127
11.5. 운영자 화면 매뉴얼	128
11.5.1. 로그인	128
11.5.2. 사용자 관리	129
11.5.3. 사용자 정보 조회	130
11.5.4. 사용자 정보 수정	132
11.5.5. 비밀번호 초기화	133
11.5.6. 사용자 등록	134

11.5.7. 그룹 관리	135
11.5.8. 그룹 등록	136
11.5.9. 그룹 정보 조회	137
11.5.10. 그룹 정보 수정	138
11.5.11. 카테고리 등록	140
11.5.12. 카테고리 정보 조회	141
11.5.13. 카테고리 정보 수정	142
11.5.14. 메뉴 관리	143
11.5.15. 메뉴 등록	144
11.5.16. 메뉴 정보 조회	145
11.5.17. 메뉴 정보 수정	146
11.5.18. 공지사항 관리	147
11.5.19. 공지사항 글쓰기	148
11.5.20. 공지사항 정보 조회	149
11.5.21. 공지사항 정보 수정	150
11.5.22. 공지사항 부가정보 추가, 삭제	151
11.5.23. 공지사항 부가정보 수정	152
11.5.24. 표준자료실 관리	153
11.5.25. 표준자료실 글쓰기	154
11.5.26. 표준자료실 정보 조회	155
11.5.27. 표준자료실 정보 수정	156
11.5.28. 표준자료실 부가정보 추가, 삭제	157
11.5.29. 표준자료실 부가정보 수정	158
11.5.30. 일반게시판 관리	159
11.5.31. 일반게시판 글쓰기	160
11.5.32. 일반게시판 정보 조회	161
11.5.33. 일반게시판 정보 수정	162
11.5.34. 일반게시판 부가정보 수정	163
11.5.35. 유관기관 링크 관리	164
11.5.36. 유관기관 링크 글쓰기	165
11.5.37. 유관기관 링크 정보 조회	166
11.5.38. 유관기관 링크 정보 수정	167
11.5.39. 유관기관 링크 부가정보 추가, 삭제	168
11.5.40. 유관기관 링크 부가정보 수정	169
11.5.41. 메타데이터 수동입력 관리	170
11.5.42. 메타데이터 수동입력 등록	171
11.5.43. 메타데이터 수동입력 정보 조회	172
11.5.44. 메타데이터 수동입력 정보 수정	174
11.5.45. 메타데이터 자동추출 관리	175
11.5.46. 메타데이터 자동추출 등록	176
11.5.47. 메타데이터 자동추출 정보 조회	177
11.5.48. 메타데이터 자동추출 정보 수정	180
11.5.49. 사용자신청 메타데이터 관리	181
11.5.50. 사용자신청 메타데이터 정보 조회	182
11.5.51. 사용자신청 메타데이터 정보 수정	183

11.5.52. 메타데이터 처리 결과 및 내용에 대한 값과 의미 184

<표 차례>

(표.1) XML Special Letter	6
(표.2) XML Type Names	8
(표.3) Reserved Words	9
(표.4) 사용자 기능 요구사항	29
(표.5) 관리자 일반 기능 요구사항	30
(표.6) 관리자 메타데이터 연동 기능 요구사항	31
(표.7) 관리자 시스템 연동 기능 요구사항	31
(표.8) 관리자 시스템 운영 통계 기능 요구사항	31
(표.9) User 테이블	36
(표.10) User Detail 테이블	36
(표.11) Group 테이블	36
(표.12) User Group 테이블	37
(표.13) Permission 테이블	37
(표.14) User Permission 테이블	37
(표.15) Group Permission 테이블	37
(표.16) Content Type 테이블	38
(표.17) Category 테이블	38
(표.18) Notice 테이블	38
(표.19) NoticeDetail 테이블	38
(표.20) Post 테이블	39
(표.21) Post Detail 테이블	39
(표.22) Common 테이블	39
(표.23) Metadata 테이블	40
(표.24) MetadataDetail 테이블	40
(표.25) MetadataRequest 테이블	40
(표.26) MetadataLog 테이블	41
(표.27) MetadataHistory 테이블	41
(표.28) 로그인	128
(표.29) 사용자 관리	129
(표.30) 사용자 정보 조회 #1	130
(표.31) 사용자 정보 조회 #2	131
(표.32) 사용자 정보 수정	132
(표.33) 비밀번호 초기화	133
(표.34) 사용자 등록	134
(표.35) 그룹 관리	135
(표.36) 그룹 등록	136
(표.37) 그룹 정보 조회	137
(표.38) 그룹 정보 수정	138
(표.39) 카테고리 관리	139
(표.40) 카테고리 등록	140
(표.41) 카테고리 정보 조회	141
(표.42) 카테고리 정보 수정	142
(표.43) 메뉴 관리	143
(표.44) 메뉴 등록	144

(표.45) 메뉴 정보 조회	145
(표.46) 메뉴 정보 수정	146
(표.47) 공지사항 관리	147
(표.48) 공지사항 글쓰기	148
(표.49) 공지사항 정보 조회	149
(표.50) 공지사항 정보 수정	150
(표.51) 공지사항 부가정보 추가,삭제	151
(표.52) 공지사항 부가정보 수정	152
(표.53) 표준자료실 관리	153
(표.54) 표준자료실 글쓰기	154
(표.55) 표준자료실 정보 조회	155
(표.56) 표준자료실 정보 수정	156
(표.57) 표준자료실 부가정보 추가,삭제	157
(표.58) 표준자료실 부가정보 수정	158
(표.59) 일반게시판 관리	159
(표.60) 일반게시판 글쓰기	160
(표.61) 일반게시판 정보 조회	161
(표.62) 일반게시판 정보 수정	162
(표.63) 일반게시판 부가정보 수정	163
(표.64) 유관기관 링크 관리	164
(표.65) 유관기관 링크 글쓰기	165
(표.66) 유관기관 링크 정보 조회	166
(표.67) 유관기관 링크 정보 수정	167
(표.68) 유관기관 링크 부가정보 추가,삭제	168
(표.69) 유관기관 링크 부가정보 수정	169
(표.70) 메타데이터 수동입력 관리	170
(표.71) 메타데이터 수동입력 등록	171
(표.72) 메타데이터 수동입력 정보 조회 #1	172
(표.73) 메타데이터 수동입력 정보 조회 #2	173
(표.74) 메타데이터 수동입력 정보 수정	174
(표.75) 메타데이터 자동추출 관리	175
(표.76) 메타데이터 자동추출 등록	176
(표.77) 메타데이터 자동추출 정보 조회 #1	177
(표.78) 메타데이터 자동추출 정보 조회 #2	178
(표.79) 메타데이터 자동추출 정보 조회 #3	179
(표.80) 메타데이터 자동추출 정보 수정	180
(표.81) 사용자신청 메타데이터 관리	181
(표.82) 사용자신청 메타데이터 정보 조회	182
(표.83) 사용자신청 메타데이터 정보 수정	183
(표.84) 메타데이터 처리 결과 및 내용	184

〈그림 차례〉

(그림.1) 컴파일러 기능	2
(그림.2) 출력 설정 기능	2
(그림.3) 컴파일러 Block Diagram	3
(그림.4) Argument Process Block	3
(그림.5) Data Registry	25
(그림.6) Emitter Block	26
(그림.7) Information Base	28
(그림.8) 메타데이터 관리시스템의 SW 블록 구조	32
(그림.9) 컨테이너 가상화 기반 메타데이터 관리시스템의 블록 구성도	33
(그림.10) 메타데이터 관리시스템의 논리적 ERD	34
(그림.11) 메타데이터 관리시스템의 물리적 ERD	35
(그림.12) 사용자 페이지 - 메인화면 상단	42
(그림.13) 사용자 페이지 - 메인화면 하단	43
(그림.14) 사용자 페이지 - 로그인 시 메인화면	43
(그림.15) 사용자 페이지 - 로그인 화면	44
(그림.16) 사용자 페이지 - 회원가입 화면	45
(그림.17) 사용자 페이지 - 공지사항 화면	45
(그림.18) 사용자 페이지 - 공지사항 게시글 화면	46
(그림.19) 사용자 페이지 - 로그인 시 공지사항 화면	46
(그림.20) 사용자 페이지 - 로그인 시 공지사항 게시글 화면	47
(그림.21) 사용자 페이지 - 로그인 시 메타데이터 조회	47
(그림.22) 사용자 페이지 - 로그인 시 메타데이터 수정/등록 신청 화면	48
(그림.23) 사용자 페이지 - 표준 자료실 화면	49
(그림.24) 사용자 페이지 - 로그인 시 표준 자료실 화면	50
(그림.25) 사용자 페이지 - 개인정보 변경 화면	50
(그림.26) 사용자 페이지 - 개인정보 활용 동의/철회 화면	51
(그림.27) 사용자 페이지 - 계정 잠금/해제 페이지	51
(그림.28) 사용자 페이지 - 비밀번호 변경 화면	52
(그림.29) 사용자 페이지 - 계정 탈퇴 화면	52
(그림.30) 관리자 페이지 - 메인화면 상단	53
(그림.31) 관리자 페이지 - 메인화면 하단	53
(그림.32) 관리자 페이지 - 로그인 화면	54
(그림.33) 관리자 페이지 - 회원가입 화면	54
(그림.34) 관리자 페이지 - 사용자 관리	55
(그림.35) 관리자 페이지 - 사용자 정보 화면 상단	55
(그림.36) 관리자 페이지 - 사용자 정보 화면 하단	56
(그림.37) 관리자 페이지 - 사용자 정보 수정 화면	56
(그림.38) 관리자 페이지 - 사용자 암호 초기화 화면	57
(그림.39) 관리자 페이지 - 그룹 관리 화면	57
(그림.40) 관리자 페이지 - 그룹 등록 화면	58
(그림.41) 관리자 페이지 - 그룹 정보 화면	58
(그림.42) 관리자 페이지 - 메인화면 관리	59
(그림.43) 관리자 페이지 - 메뉴 관리	59
(그림.44) 관리자 페이지 - 메뉴 등록 화면	60

(그림.45) 관리자 페이지 - 배너 이미지 수정/등록	60
(그림.46) 관리자 페이지 - 공지사항 수정/등록	61
(그림.47) 관리자 페이지 - 표준 자료실 수정/등록	61
(그림.48) 관리자 페이지 - 관련 사이트 링크 수정/등록	62
(그림.49) 관리자 페이지 - 유관기관 링크 수정/등록	62
(그림.50) 관리자 페이지 - 공지사항 관리	63
(그림.51) 관리자 페이지 - 공지사항 정보	63
(그림.52) 관리자 페이지 - 공지사항 수정/등록	64
(그림.53) 관리자 페이지 - 표준 자료실 관리	64
(그림.54) 관리자 페이지 - 표준 자료 정보	65
(그림.55) 관리자 페이지 - 표준 자료 수정/등록	65
(그림.56) 관리자 페이지 - 카테고리 관리	66
(그림.57) 관리자 페이지 - 카테고리 등록	66
(그림.58) 관리자 페이지 - 카테고리 정보	67
(그림.59) 관리자 페이지 - 메타데이터 관리	67
(그림.60) 관리자 페이지 - 메타데이터 입/출력	68
(그림.61) 관리자 페이지 - 메타데이터 수정 신청	68
(그림.62) 사이트맵 - 사용자 페이지	69
(그림.63) 사이트맵 - 관리자 페이지_사용자/그룹	69
(그림.64) 사이트맵 - 관리자 페이지_디스플레이	70
(그림.65) python django 의 Model, View, Template 역할과 상호 연동 방법	71
(그림.66) 메타데이터 관리시스템과 전용컴파일러 간 호출 및 데이터 연동 방식	72
(그림.67) 정형 데이터의 테이블 저장 방식	73
(그림.68) 반정형 데이터의 테이블 저장 방식	74
(그림.69) Interface Module	76
(그림.70) Interface Module Host	76
(그림.71) Interface Module Servant	76
(그림.72) Host Module Initialize	77
(그림.73) Servant Module Initialize	77
(그림.74) Establishment & Negotiation	77
(그림.75) Polling	79
(그림.76) Functionality Invocation	79
(그림.77) IPC Frame	80
(그림.78) Protocol Engine	83
(그림.79) Message Broker	83
(그림.80) Protocol Agent	83
(그림.81) CODEC Engine	84
(그림.82) 초기화 기능	85
(그림.83) Message Broker 초기화 기능	85
(그림.84) Message Routing 기능	85
(그림.85) External Asset 연동 기능	86
(그림.86) Encapsulation	86
(그림.87) Decapsulation	86
(그림.88) Finite State Machine	87
(그림.89) Protocol Agent Control	87

(그림.90) Encoder	87
(그림.91) Decoder	88
(그림.92) 시나리오엔진 연동시스템 내부구조	91
(그림.93) websocket 접속 절차	93
(그림.94) websocket 헤더와 페이로드 구조	93
(그림.95) 시나리오엔진과 표준적합성 검증시스템과의 연동 절차	95
(그림.96) 표준적합성 검증시스템의 내부 구조	96
(그림.97) nodejs 의 내부 구조	98
(그림.98) websocket 접속 절차	98
(그림.99) websocket 헤더와 페이로드 구조	99
(그림.100) 표준적합성 검증시스템과 시나리오엔진 연동시스템의 연동 절차	101
(그림.101) 메타데이터 관리시스템의 운영 환경	118
(그림.102) 메타데이터 관리시스템 내부구조	119
(그림.103) 메타데이터 관리시스템과 전용 컴파일러 연동 구조	120

1. 개요

1.1. 과업 개요

- 본 과업에서는 자율주행 혼재 시 도로교통 통합관계센터와 교통안전시설 인프라(교통안전 보조시설 포함), 자율주행차 등 정보연계 대상 간 표준 준용 여부 검증에 필요한 표준검증플랫폼 세부 시스템 및 구현 기능을 설계한다.
- 통합관계시스템에 연동되는 시스템들의 표준 준용을 통해 전체 플랫폼의 안정성을 보장할 수 있으며, 표준 적합성 테스트 환경을 구축하고 범용적으로 표준적합성 테스트를 수행할 수 있는 여건을 마련하여 공공·민간 사업자가 표준 준용에 대한 부담과 시스템 개발에 필요한 공수를 절감할 수 있게 하는 기대효과를 볼 수 있다.
- 본 최종보고서는 표준검증플랫폼 시스템 개발 및 설계에 대한 항목들과 표준검증플랫폼 이용에 대한 사용자/운영자 매뉴얼 등을 포함한다.

1.2. 설계 항목

- 표준검증플랫폼의 각 설계 항목 및 주요 기능은 아래 표와 같다.

연번	설계 범위	주요 기능
1	전용 컴파일러	화면에서 입력된 자료구조 표준문서(ANS.1 등)를 파싱(parsing)하여 표준 적합성 검증에서 사용할 메타데이터를 추출 메타데이터 DB에 검증플랫폼 코드를 제공
2	메타데이터 DB	전용 컴파일러에서 추출된 메타데이터와 메타데이터 관리항목이 적재된 데이터베이스
3	메타데이터 관리시스템	추출된 메타데이터 DB 적재 및 데이터 관리 메타데이터 조회, 등록, 변경, 삭제 기능을 구현한 WEB 기반 관리시스템
4	시나리오 엔진	다양한 프로토콜에 대한 표준적합성 테스트 시나리오 관련 표준 분석 메타데이터를 이용해 자료구조 인코딩/디코딩 수행 표준에 제시된 테스트 시나리오 기반으로 검증할 수 있도록 메시지 구성 송수신 메시지 비교 및 검증하여 표준적합성 준용 여부 확인 (입력/출력 변수에 표준 범위 내의 값이 할당되도록 구성) 시나리오 엔진연동시스템은 시나리오 엔진과 웹페이지(표준적합성 검증시스템)와 연동을 위한 기능 수행
5	인터페이스	정보연계 검증 대상의 다양한 프로토콜에 맞게 변환하여 연동 수행 시나리오에서 작성된 메시지 송수신
6	표준적합성 검증시스템 (WEB)	검증대상의 표준 준용 여부 테스트 결과를 출력하는 WEB 기반 표준적합성 검증시스템으로 메타데이터 관리시스템과 연계하여 하나의 WEB 화면으로 표출

2. 전용 컴파일러

ASN.1 전용 컴파일러는 ITU-T X.680, X.681, X.682 및 X.683을 준용하여 개발한다. 단, 추상 구문의 표현을 완벽하게 LR Parser로 변형하는 것은 한계가 있으므로, 각 표준을 100% 준수하는 것은 불가능하다는 점을 명시한다.

ASN.1 표준에 정의한 Lexical Item과 Syntax를 분석하고, 구문 분석에 의해 수집된 정보를 저장할 구조를 설계한다. 저장 구조체를 이용하여 출력하기 위한 블록을 설계한다. 출력은 다양한 형태의 Scheme을 지원하도록 설계하여, 컴파일러의 활용성을 극대화 한다.

2.1. 기능 설계

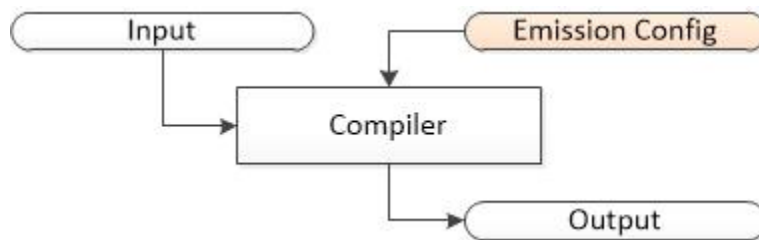
컴파일러는 입력을 받아 특정 출력을 내는 것을 의미하므로 출력에 대한 기능에 집중하여 설계한다.



(그림.1) 컴파일러 기능

2.2. 출력 설정 기능

전용 컴파일러는 ASN.1 문서로부터 메타데이터 관리 시스템에 공급할 세부 Data와 검증 시스템에 공급할 CODEC Data를 추출하여야 한다. 본 요구사항을 만족시키기 위해, 출력 서식을 Runtime에 결정할 수 있도록 출력 설정 기능을 설계 한다.



(그림.2) 출력 설정 기능

Emission Config은 다음과 같은 정보를 제공하도록 설계한다.

- Header

Header는 입력 ASN.1 문서에서 수집된 전반적인 정보를 어떻게 출력할지 정의한다.

- Elements

Elements는 ASN.1 문서에서 수집된 각 Type에 대한 정보의 List이다.

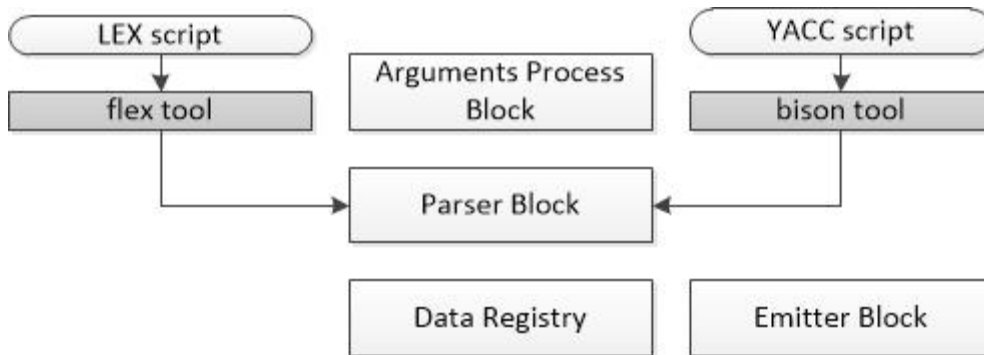
- Element

Element는 하나의 Type에 대한 정보를 어떻게 출력할지 정의한다.

Compiler는 제공된 Emission Config에 따라 정보를 출력하도록 설계한다.

2.3. 블록 설계

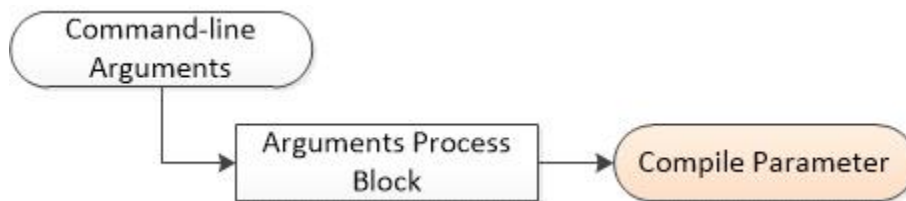
전용 컴파일러는 Argument Process Block, Parser Block, Data Registry, Emitter Block으로 구성된다.



(그림.3) 컴파일러 Block Diagram

2.3.1. Arguments Process Block

Arguments Process Block은 인수로 전달된 정보를 처리하는 역할을 담당한다.



(그림.4) Argument Process Block

Command-line Arguments를 분석하여 생성되는 Compile Parameter는 다음과 같다.

- ASN.1 Document Files

입력에 사용할 ASN.1 File 목록

- Emission Configuration File

Emission Config이 저장된 File의 경로 및 파일명

- Inclusion Registry Path

Import 구문 처리를 위한 ASN.1 Registry 경로

2.3.2. Parser Block

Parser Block은 compiler 제작 tool인 flex와 bison을 통해 구현한다. flex와 bison은 각각 입력 script를 요구하기 때문에 각 script를 작성하기 위한 lexical item 및 syntax를 분석한다.

2.3.2.1. Lexical item

Lexical item은 특정 문자열 패턴에 부합하는 문자열을 Token으로 인식되는 대상으로 X.680 및 X.681에 정의된 Lexical item은 다음과 같다.

- typereference

문자와 숫자, 하이픈으로 구성되며 반드시 첫문자는 대문자로 시작하여야 한다. 하이픈은 연속하여 사용할 수 없다. typereference에 사용되는 문자열은 예약어와 겹치면 안된다.

- identifiers

문자와 숫자, 하이픈으로 구성되며 반드시 첫문자는 소문자로 시작하여야 한다. 하이픈은 연속하여 사용할 수 없다.

- valuereference

identifier와 동일한 패턴을 사용한다. 다만 identifier와 의미적 구별을 두기위해 사용한다. valuereference는 다른 identifier를 참조할 때 사용한다.

- modulereference

identifier와 동일한 패턴을 사용한다. 다만 identifier와 의미적 구별을 두기 위해 사용한다. modulereference는 다른 module을 참조할 때 사용한다.

- comment

주석은 2가지 패턴을 사용한다.

두개의 하이픈("--")으로 시작하여 개행 또는 두 개의 하이픈("--")에서 종료
SLASH와 ASTERISK("/>")로 시작하여 ASTERISK와 SLASH로 종료("*/")

- empty

empty는 아무 패턴을 지정하지 않는다. 이 item은 구문 분석이 종료될 수 있는 여지를 제공하기위해 사용한다.

- number

숫자로 구성되며, 0을 표현하지 않는 이상 0으로 시작할 수 없다.

- realnumber

지수, 가수 승수로 구성되며, 지수와 가수 가이에 FULL-STOP(.)이 있어야 한다. 승수는 반드시 E로 시작하여야 한다. 지수, 가수, 승수는 모두 number의 규칙을 사용하며, 지수와 승수는 HYPHEN-MINUS(-)로 시작 할 수 있다.

- bstring

APOSTROPHE(')로 시작하고 APOSTROPHE(')와 B로 끝나야 한다. APOSTROPHE(') 사이에는 0과 1만 사용할 수 있으며, white-space를 사용할 수 있지만 이는 무시된다.

- xmlbstring

0과 1만 사용할 수 있으며, white-space를 사용할 수 있지만 이는 무시된다. (시작과 끝은 XML Tag를 이용하여 알 수 있다.)

- hstring

APOSTROPHE(')로 시작하고 APOSTROPHE(')와 H로 끝나야 한다. APOSTROPHE(') 사이에는 0에서 9, A에서 F만 사용할 수 있으며, white-space를 사용할 수 있지만 이는 무시된다.

- xmlhstring

0에서 9, A에서 F만 사용할 수 있으며, white-space를 사용할 수 있지만 이는 무시된다. (시작과 끝은 XML Tag를 이용하여 알 수 있다.)

- cstring

QUOTATION-MARK(“) 사이에 어떤 문자든지 사용할 수 있다. 문자열 내에 QUOTATION-MARK(“)를 포함 시키고자 할 때 white-space없이 연속하여 두 번(”) 사용한다.

- xmlcstring

ISO/IEC 10636 표준에 명시한 문자 중 다음에 열거한 문자만 사용한다.

HORIZONTAL TABULATION(9)

LINE FEED(10)

CARRIAGE RETURN(13)

0x20 ~ 0xD7FF 영역의 문자

0xE000 ~ 0xFFFFD 영역의 문자

0x10000 ~ 0x10FFFF 영역의 문자

일부 문자는 다음과 같이 표현하여야 한다.

AMPERSAND(&) = &

LESS-THAN SIGN(<) = <

- GREATER-THAN SIGN(>) = >

0x00~0x1F 영역의 문자를 표현할 때는 다음 표에 따른다.

(표.1) XML Special Letter

ISO/IEC 10646 code	X M L	ISO/IEC 10646 code	X M L
0x00	<null/>	0x10	<delete/>
0x01	<single/>	0x11	<delete1/>
0x02	<single2/>	0x12	<delete2/>
0x03	<single3/>	0x13	<delete3/>
0x04	<single4/>	0x14	<delete4/>
0x05	<singleq/>	0x15	<singlequote/>
0x06	<singlek/>	0x16	<singleyn/>
0x07	<singlel/>	0x17	<singleetb/>
0x08	<singlebs/>	0x18	<singlecan/>
0x09		0x19	<singleem/>
0x0A		0x1A	<singlesub/>
0x0B	<singlevt/>	0x1B	<singleesc/>
0x0C	<singleff/>	0x1C	<singleis4/>
0x0D		0x1D	<singleis3/>
0x0E	<singleso/>	0x1E	<singleis2/>
0x0F	<singlesi/>	0x1F	<singleis1/>

편집기의 한계로 입력할 수 없는 문자는 다음과 같은 방법으로 작성한다.

&#N: N은 10진수 code

&#xH: H는 16진수 code

- simplestring

cstring과 동일. Time Type의 subtype을 지정하는 용도로 구별하기 위해 사용

- tstring

QUOTATION-MARK(“) 사이에 다음에 열거한 문자만 허용

0 1 2 3 4 5 6 7 8 9 + - : . , / C D H M R P S T W Y Z

시간값을 지정하기 위해 사용

- xmltstring
tstring과 동일. XML 구문상에서 사용하는 용도로 구별하기 위해 사용
- psname
property and setting name의 약어로, 문자, 숫자와 HYPHEN(-)을 사용하여 구성. HYPHEN(-)은 마지막에 올 수 없으며 두 번이상 연속될 수 없음.
Tile Type의 subtype을 지정하는 용도로 구별하기 위해 사용
- encodingreference
typereference의 정의를 기본으로, 모두 대문자만 사용하여 구성. 인코딩 방법을 설정하기위한 용도로 구별하기 위해 사용.
- integerUnicodeLabel
number와 동일. OID의 arc값을 지정하기 위한 용도로 구별하기 위해 사용.
non-integerUnicodeLabel
ISO/IEC 10646의 7.2.5에 정의된 제약조건을 만족하는 문자만 사용하여 구성. OID의 arc값을 지정하기 위한 용도로 구별하기 위해 사용.
- objectclassreference
typereference의 정의를 기본으로, 모두 대문자만 사용하여 구성. CLASS를 지정하기 위한 용도로 구별하기 위해 사용.
- objectreference
valuereference와 동일. object를 지정하기 위한 용도로 구별하기 위해 사용.
- objectsetreference
typereference와 동일. object set을 지정하기 위한 용도로 구별하기 위해 사용.
- typefieldreference
typereference의 정의를 기본으로, AMPHERSAND(&)로 시작. Class의 Field를 지정하기 위한 용도로 구별하기 위해 사용.
- valuefieldreference
valuereference의 정의를 기본으로, AMPHERSAND(&)로 시작. Information object의 값을 지정하기 위한 용도로 구별하기 위해 사용.
- valuesetfieldreference
typereference의 정의를 기본으로, AMPHERSAND(&)로 시작. Information Object Set의 값을 지정하기 위한 용도로 구별하기 위해 사용.
- objectfieldreference
objectreference의 정의를 기본으로, AMPHERSAND(&)로 시작. object의 field를 지정하기 위한 용도로 구별하기 위해 사용.
- objectsetfieldreference
objectsetreference의 정의를 기본으로, AMPHERSAND(&)로 시작. object set의 field를 지정하기 위한 용도로 구별하기 위해 사용.
- word
typereference의 정의를 기본으로, 소문자와 숫자를 허용하지 않음. WITH SYNTAX구문의 Token을 정위하기 위한 용도로 구별하기 위해 사용.
- XML Type Names
XML Tag에 나타나는 이름중 다음에 열거한 이름은 해당 이름에 대한 내장 Type으로 치환하여 사용해야 함.

(표.2) XML Type Names

ASN.1 type production name	Characters in xmlasn1typename
BitStringType	BIT_STRING
BooleanType	BOOLEAN
ChoiceType	CHOICE
DateType	DATE
DateTimeType	DATE_TIME
DurationType	DURATION
EmbeddedPDVType	SEQUENCE
EnumeratedType	ENUMERATED
ExternalType	SEQUENCE
InstanceOfType	SEQUENCE
IntegerType	INTEGER
IRIType	OID_IRI
NullType	NULL
ObjectClassFieldType	See ITU-T X.681 14.10 and 14.11
ObjectIdentifierType	OBJECT_IDENTIFIER
OctetStringType	OCTET_STRING
PrefixedType	See 12.36.5
RealType	REAL
RelativeIRIType	RELATIVE_OID_IRI
RelativeOIDType	RELATIVE_OID
RestrictedCharacterStringType	The type name (e.g. IA5String)
SequenceType	SEQUENCE
SequenceOfType	SEQUENCE_OF
SetType	SET
SetOfType	SET_OF
TimeType	TIME
TimeOfDayType	TIME_OF_DAY
UnrestrictedCharacterStringType	SEQUENCE

- Single character

다음에 열거한 단일 문자는 각각의 개별 토큰으로 인식되어 구문 분석에 사용.

"{", "}", "<", ">", ",", ".", "/", "(", ")", "[", "]", "-", "(HYPHEN-MINUS), ":", "=", ""(QUOTATION MARK), ""(APOSTROPHE), " "(SPACE), ";", "@", "|", "!", "^",

- Letter Combinations

다음에 열거한 문자열은 하나의 토큰으로 인식하며, 구문 분석에 특별한 용도로 사용

“::=”, “..”, “...”, “[[”, “[]”

- XML-Related

다음에 열거한 문자열은 하나의 토큰으로 인식하며, XML 구문 분석에 특별한 용도로 사용.

“</”, “/>”, “NaN”, “INF”,

- Reserved words

다음에 열거한 예약어는 Lexical item에 사용할 수 없음.

(표.3) Reserved Words

ABSENT	ENCODED	INTERSECTION	SEQUENCE
ABSTRACT-SYN	ENCODING-CON	ISO646String	SET
TAX	TROL		
ALL	END	MAX	SETTINGS
APPLICATION	ENUMERATED	MIN	SIZE
AUTOMATIC	EXCEPT	MINUS-INFINITY	STRING
BEGIN	EXPLICIT	NOT-A-NUMBER	SYNTAX
BIT	EXPORTS	NULL	T61String
BMPString	EXTENSIBILITY	NumericString	TAGS
BOOLEAN	EXTERNAL	OBJECT	TeletexString
BY	FALSE	ObjectDescriptor	TIME
CHARACTER	FROM	OCTET	TIME-OF-DAY
CHOICE	GeneralizedTime	OF	TRUE
CLASS	GeneralString	OID-IRI	TYPE-IDENTIFIE
			R
COMPONENT	GraphicString	OPTIONAL	UNION
COMPONENTS	IA5String	PATTERN	UNIQUE
CONSTRAINED	IDENTIFIER	PDV	UNIVERSAL
CONTAINING	IMPLICIT	PLUS-INFINITY	UniversalString
DATE	IMPLIED	PRESENT	UTCTime
DATE-TIME	IMPORTS	PrintableString	UTF8String
DEFAULT	INCLUDES	PRIVATE	VideotexString
DEFINITIONS	INSTANCE	REAL	VisibleString
DURATION	INSTRUCTIONS	RELATIVE-OID	WITH
EMBEDDED	INTEGER	RELATIVE-OID-I	
		RI	

2.3.2.2. 기타

다음 항목들은 Lexical item에 포함되지 않는 경우 무시됨.

- wite-space

HORIZONTAL TABULATION(9)

LINE FEED(10)

VERTICAL TABULATION(11)

FORM FEED(12)

CARRIAGE RETURN(13)

SPACE(32)

NO-BREAK SPACE({0,0,0,160})

- new-line

LINE FEED(10)

VERTICAL TABULATION(11)

FORM FEED(12)

CARRIAGE RETURN(13)

2.3.3. Syntax

ASN.1의 Syntax는 X.680에서 X.683에 걸쳐 정의하고 있으며, 다음과 같음

2.3.3.1. X.680

```

ModuleDefinition ::=
    ModuleIdentifier
    DEFINITIONS
    EncodingReferenceDefault

```

```

    TagDefault
    ExtensionDefault
    " := "
    BEGIN
    ModuleBody
    EncodingControlSections
    END

ModuleIdentifier ::= modulereference DefinitiveIdentification
DefinitiveIdentification ::= DefinitiveOID | DefinitiveOIDandIRI | empty
DefinitiveOID ::= "{" DefinitiveObjIdComponentList "}"
DefinitiveOIDandIRI ::= DefinitiveOID IRIRValue
DefinitiveObjIdComponentList ::=
    DefinitiveObjIdComponent
    | DefinitiveObjIdComponent DefinitiveObjIdComponentList
DefinitiveObjIdComponent ::=
    NameForm
    | DefinitiveNumberForm
    | DefinitiveNameAndNumberForm
DefinitiveNumberForm ::= number
DefinitiveNameAndNumberForm ::= identifier "(" DefinitiveNumberForm ")"
EncodingReferenceDefault ::=
    encodingreference INSTRUCTIONS
    | empty
TagDefault ::=
    EXPLICIT TAGS
    | IMPLICIT TAGS
    | AUTOMATIC TAGS
    | empty
ExtensionDefault ::=
    EXTENSIBILITY IMPLIED
    | empty
ModuleBody ::=
    Exports Imports AssignmentList
    | empty
Exports ::=
    EXPORTS SymbolsExported ";"
    | EXPORTS ALL ";"
    | empty
SymbolsExported ::= SymbolList | empty
Imports ::=
    IMPORTS SymbolsImported ";"
    | empty
SymbolsImported ::= SymbolsFromModuleList | empty
SymbolsFromModuleList ::=
    SymbolsFromModule
    | SymbolsFromModuleList SymbolsFromModule
SymbolsFromModule ::= SymbolList FROM GlobalModuleReference
GlobalModuleReference ::= modulereference AssignedIdentifier
AssignedIdentifier ::= ObjectIdentifierValue | DefinedValue | empty
SymbolList ::=
    Symbol
    | SymbolList ";" Symbol

```


Symbol ::= Reference | ParameterizedReference

Reference ::=

- typereference
- | valuereference
- | objectclassreference
- | objectreference
- | objectsetreference

AssignmentList ::=

- Assignment
- | AssignmentList Assignment

Assignment ::=

- TypeAssignment
- | ValueAssignment
- | XMLValueAssignment
- | ValueSetTypeAssignment
- | ObjectClassAssignment
- | ObjectAssignment
- | ObjectSetAssignment
- | ParameterizedAssignment

DefinedType ::=

- ExternalTypeReference
- | typereference
- | ParameterizedType
- | ParameterizedValueSetType

DefinedValue ::= ExternalValueReference | valuereference | ParameterizedValue

NonParameterizedTypeName ::= ExternalTypeReference | typereference | xmlnsn1typename

ExternalTypeReference ::= modulereference "." typereference

ExternalValueReference ::= modulereference "." valuereference

AbsoluteReference ::= "@" ModuleIdentifier "." ItemSpec

ItemSpec ::=

- typereference
- | ItemId "." ComponentId

ItemId ::= ItemSpec

ComponentId ::= identifier | number | "*"

TypeAssignment ::= typereference ": :=" Type

ValueAssignment ::= valuereference Type ": :=" Value

XMLValueAssignment ::= valuereference ": :=" XMLTypedValue

XMLTypedValue ::=

- "<" & NonParameterizedTypeName ">"
- XMLValue
- "</" & NonParameterizedTypeName ">"
- | "<" & NonParameterizedTypeName "/>"

ValueSetTypeAssignment ::=

- typereference Type ": :=" ValueSet

ValueSet ::= "{" ElementSetSpecs "}"

Type ::= BuiltinType | ReferencedType | ConstrainedType

BuiltinType ::=

- BitStringType
- | BooleanType
- | CharacterStringType
- | ChoiceType
- | DateType
- | DateTimeType

```

| DurationType
| EmbeddedPDVType
| EnumeratedType
| ExternalType
| InstanceOfType
| IntegerType
| IRType
| NullType
| ObjectClassFieldType
| ObjectIdentifierType
| OctetStringType
| RealType
| RelativeIRType
| RelativeOIDType
| SequenceType
| SequenceOfType
| SetType
| SetOfType
| PrefixedType
| TimeType
| TimeOfDayType

ReferencedType ::=
    DefinedType
    | UsefulType
    | SelectionType
    | TypeFromObject
    | ValueSetFromObjects

NamedType ::= identifier Type

Value ::= BuiltinValue | ReferencedValue | ObjectClassFieldValue

XMLValue ::= XMLBuiltinValue | XMLObjectClassFieldValue

BuiltinValue ::=
    BitStringValue
    | BooleanValue
    | CharacterStringValue
    | ChoiceValue
    | EmbeddedPDVValue
    | EnumeratedValue
    | ExternalValue
    | InstanceOfValue
    | IntegerValue
    | IRValue
    | NullValue
    | ObjectIdentifierValue
    | OctetStringValue
    | RealValue
    | RelativeIRValue
    | RelativeOIDValue
    | SequenceValue
    | SequenceOfValue
    | SetValue
    | SetOfValue
    | PrefixedValue
    | TimeValue

XMLBuiltinValue ::=
    XMLBitStringValue
    | XMLBooleanValue
    | XMLCharacterStringValue
    | XMLChoiceValue
    | XMLEmbeddedPDVValue
    | XMLEnumeratedValue
    | XMLExternalValue
    | XMLInstanceOfValue

```

```

| XMLIntegerValue
| XMLIRValue
| XMLNullValue
| XMLObjectIdentifierValue
| XMLOctetStringValue
| XMLRealValue
| XMLRelativeIRValue
| XMLRelativeOIDValue
| XMLSequenceValue
| XMLSequenceOfValue
| XMLSetValue
| XMLSetOfValue
| XMLPrefixedValue
| XMLTimeValue

```

ReferencedValue ::= DefinedValue | ValueFromObject

NamedValue ::= identifier Value

XMLNamedValue ::= "<" & identifier ">" XMLValue "</" & identifier ">"

BooleanType ::= BOOLEAN

BooleanValue ::= TRUE | FALSE

XMLBooleanValue ::= EmptyElementBoolean | TextBoolean

```

EmptyElementBoolean ::=
    "<" & "true" ">"
|
    "<" & "false" ">"

```

TextBoolean ::= extended-true | extended-false

```

IntegerType ::=
    INTEGER
|
    INTEGER "{" NamedNumberList "}"

```

```

NamedNumberList ::=
    NamedNumber
|
    NamedNumberList "," NamedNumber

```

```

NamedNumber ::=
    identifier "(" SignedNumber ")"
|
    identifier "(" DefinedValue ")"

```

```

SignedNumber ::=
    number
|
    "-" number

```

IntegerValue ::= SignedNumber | identifier

```

XMLIntegerValue ::=
    XMLSignedNumber
|
    EmptyElementInteger
|
    TextInteger

```

```

XMLSignedNumber ::=
    number
|
    "-" & number

```

EmptyElementInteger ::= "<" & identifier ">"

TextInteger ::= identifier

EnumeratedType ::= ENUMERATED "{" Enumerations "}"

```

Enumerations ::=
    RootEnumeration
|
    RootEnumeration ";" "..." ExceptionSpec
|
    RootEnumeration ";" "..." ExceptionSpec ";" AdditionalEnumeration

```

RootEnumeration ::= Enumeration

AdditionalEnumeration ::= Enumeration

```

Enumeration ::=
    EnumerationItem
    | EnumerationItem "," Enumeration
EnumerationItem ::= identifier | NamedNumber
EnumeratedValue ::= identifier
XMLEnumeratedValue ::= EmptyElementEnumerated | TextEnumerated
EmptyElementEnumerated ::= "<" & identifier ">"
TextEnumerated ::= identifier
RealType ::= REAL
RealValue ::= NumericRealValue | SpecialRealValue
NumericRealValue ::=
    realnumber
    | "-" realnumber
    | SequenceValue
SpecialRealValue ::= PLUS-INFINITY | MINUS-INFINITY | NOT-A-NUMBER
XMLRealValue ::= XMLNumericRealValue | XMLSpecialRealValue
XMLNumericRealValue ::=
    realnumber
    | "-" & realnumber
XMLSpecialRealValue ::= EmptyElementReal | TextReal
EmptyElementReal ::=
    "<" & PLUS-INFINITY ">"
    | "<" & MINUS-INFINITY ">"
    | "<" & NOT-A-NUMBER ">"
TextReal ::=
    "INF"
    | "-" & "INF"
    | "NaN"
BitStringType ::=
    BIT STRING
    | BIT STRING "{" NamedBitList "}"
NamedBitList ::=
    NamedBit
    | NamedBitList "," NamedBit
NamedBit ::=
    identifier "(" number ")"
    | identifier "(" DefinedValue ")"
BitStringValue ::=
    bstring
    | hstring
    | "{" IdentifierList "}"
    | "{" "}"
    | CONTAINING Value
IdentifierList ::=
    identifier
    | IdentifierList "," identifier
XMLBitStringValue ::=
    XMLTypedValue
    | xmlbstring
    | XMLIdentifierList
    | empty

```

```

XMLIdentifierList ::=
    EmptyElementList
    | TextList
EmptyElementList ::=
    "<" & identifier ">"
    | EmptyElementList "<" & identifier ">"
TextList ::=
    identifier
    | TextList identifier
OctetStringType ::= OCTET STRING
OctetStringValue ::=
    bstring
    | hstring
    | CONTAINING Value
XMLOctetStringValue ::=
    XMLTypedValue
    | xmlhstring
NullType ::= NULL
NullValue ::= NULL
XMLNullValue ::= empty
SequenceType ::=
    SEQUENCE "{" "}"
    | SEQUENCE "{" ExtensionAndException OptionalExtensionMarker "}"
    | SEQUENCE "{" ComponentTypeLists "}"
ExtensionAndException ::=
    "..."
    | "..." ExceptionSpec
OptionalExtensionMarker ::=
    ";" "..."
    | empty
ComponentTypeLists ::=
    RootComponentTypeList
    | RootComponentTypeList
      " "
      ExtensionAndException
      ExtensionAdditions
      OptionalExtensionMarker
    | RootComponentTypeList
      " / "
      ExtensionAndException
      ExtensionAdditions
      ExtensionEndMarker
      " / "
      RootComponentTypeList
    | ExtensionAndException
      ExtensionAdditions
      ExtensionEndMarker
      " / "
      RootComponentTypeList
    | ExtensionAndException
      ExtensionAdditions
      OptionalExtensionMarker
RootComponentTypeList ::=
    ComponentTypeList
ExtensionEndMarker ::= ";" "..."

```

```

ExtensionAdditions ::=
    ";" ExtensionAdditionList
    |
    empty

ExtensionAdditionList ::=
    ExtensionAddition
    |
    ExtensionAdditionList ";" ExtensionAddition

ExtensionAddition ::=
    ComponentType
    |
    ExtensionAdditionGroup

ExtensionAdditionGroup ::=
    "[[" VersionNumber ComponentTypeList "]" ]]"

VersionNumber ::=
    empty
    |
    number ":"

ComponentTypeList ::=
    ComponentType
    |
    ComponentTypeList ";" ComponentType

ComponentType ::=
    NamedType
    |
    NamedType OPTIONAL
    |
    NamedType DEFAULT Value
    |
    COMPONENTS OF Type

SequenceValue ::=
    "{" ComponentValueList "}"
    |
    "{" "}"

ComponentValueList ::=
    NamedValue
    |
    ComponentValueList ";" NamedValue

XMLSequenceValue ::=
    XMLComponentValueList
    |
    empty

XMLComponentValueList ::=
    XMLNamedValue
    |
    XMLComponentValueList XMLNamedValue

SequenceOfType ::=
    SEQUENCE OF Type
    |
    SEQUENCE OF NamedType

SequenceOfValue ::=
    "{" ValueList "}"
    |
    "{" NamedValueList "}"
    |
    "{" "}"

ValueList ::=
    Value
    |
    ValueList ";" Value

NamedValueList ::=
    NamedValue
    |
    NamedValueList ";" NamedValue

XMLSequenceOfValue ::=
    XMLValueList
    |
    XMLDelimitedItemList
    |
    empty

XMLValueList ::=
    XMLValueOrEmpty
    |
    XMLValueOrEmpty XMLValueList

```

```

XMLValueOrEmpty ::=
    XMLValue
    | "<" & NonParameterizedTypeName ">"

XMLDelimitedItemList ::=
    XMLDelimitedItem
    | XMLDelimitedItem XMLDelimitedItemList

XMLDelimitedItem ::=
    "<" & NonParameterizedTypeName ">"
    XMLValue
    "</" & NonParameterizedTypeName ">"
    | "<" & identifier ">" XMLValue "</" & identifier ">"

SetType ::=
    SET "{" "}"
    | SET "{" ExtensionAndException OptionalExtensionMarker "}"
    | SET "{" ComponentTypeLists "}"

SetValue ::=
    "{" ComponentValueList "}"
    | "{" "}"

XMLSetValue ::=
    XMLComponentValueList
    | empty

SetOfType ::=
    SET OF Type
    | SET OF NamedType

SetOfValue ::=
    "{" ValueList "}"
    | "{" NamedValueList "}"
    | "{" "}"

XMLSetOfValue ::=
    XMLValueList
    | XMLDelimitedItemList
    | empty

ChoiceType ::= CHOICE "{" AlternativeTypeLists "}"

AlternativeTypeLists ::=
    RootAlternativeTypeList
    | RootAlternativeTypeList
    |
    | ExtensionAndException
    | ExtensionAdditionAlternatives
    | OptionalExtensionMarker

RootAlternativeTypeList ::= AlternativeTypeList

ExtensionAdditionAlternatives ::=
    ";" ExtensionAdditionAlternativesList
    | empty

ExtensionAdditionAlternativesList ::=
    ExtensionAdditionAlternative
    | ExtensionAdditionAlternativesList ";" ExtensionAdditionAlternative

ExtensionAdditionAlternative ::=
    ExtensionAdditionAlternativesGroup
    | NamedType

ExtensionAdditionAlternativesGroup ::= "[" VersionNumber AlternativeTypeList "]"

AlternativeTypeList ::=
    NamedType
    | AlternativeTypeList ";" NamedType

```

ChoiceValue ::= identifier ":" Value
XMLChoiceValue ::= "<" & identifier ">" XMLValue "</" & identifier ">"
SelectionType ::= identifier "<" Type
PrefixedType ::=
 TaggedType
 | EncodingPrefixedType
PrefixedValue ::= Value
XMLPrefixedValue ::= XMLValue
EncodingPrefixedType ::= EncodingPrefix Type
EncodingPrefix ::= "[" EncodingReference EncodingInstruction "]"
TaggedType ::=
 Tag Type
 | Tag IMPLICIT Type
 | Tag EXPLICIT Type
Tag ::= "[" EncodingReference Class ClassNumber "]"
EncodingReference ::=
 encodingreference ":"
 | empty
ClassNumber ::=
 number
 | DefinedValue
Class ::=
 UNIVERSAL
 | APPLICATION
 | PRIVATE
 | empty
EncodingPrefixedType ::= EncodingPrefix Type
EncodingPrefix ::= "[" EncodingReference EncodingInstruction "]"
ObjectIdentifierType ::= OBJECT IDENTIFIER
ObjectIdentifierValue ::=
 {" ObjIdComponentsList }"
 | {" DefinedValue ObjIdComponentsList }"
ObjIdComponentsList ::=
 ObjIdComponents
 | ObjIdComponents ObjIdComponentsList
ObjIdComponents ::=
 NameForm
 | NumberForm
 | NameAndNumberForm
 | DefinedValue
NameForm ::= identifier
NumberForm ::= number | DefinedValue
NameAndNumberForm ::= identifier "(" NumberForm ")"
XMLObjectIdentifierValue ::=
 XMLObjIdComponentList
XMLObjIdComponentList ::=
 XMLObjIdComponent
 | XMLObjIdComponent & "." & XMLObjIdComponentList
XMLObjIdComponent ::=
 NameForm


```

    | XMLNumberForm
    | XMLNameAndNumberForm
XMLNumberForm ::= number
XMLNameAndNumberForm ::=
    identifier & "(" & XMLNumberForm & ")"
RelativeOIDType ::= RELATIVE-OID
RelativeOIDValue ::=
    "{" RelativeOIDComponentsList "}"
RelativeOIDComponentsList ::=
    RelativeOIDComponents
    | RelativeOIDComponents RelativeOIDComponentsList
RelativeOIDComponents ::=
    NumberForm
    | NameAndNumberForm
    | DefinedValue
XMLRelativeOIDValue ::=
    XMLRelativeOIDComponentList
XMLRelativeOIDComponentList ::=
    XMLRelativeOIDComponent
    | XMLRelativeOIDComponent & "." & XMLRelativeOIDComponentList
XMLRelativeOIDComponent ::=
    XMLNumberForm
    | XMLNameAndNumberForm
IRIType ::= OID-IRI
IRIValue ::=
    "" FirstArcIdentifier SubsequentArcIdentifier ""
FirstArcIdentifier ::=
    "/" ArcIdentifier
SubsequentArcIdentifier ::=
    "/" ArcIdentifier SubsequentArcIdentifier
    | empty
ArcIdentifier ::=
    integerUnicodeLabel
    | non-integerUnicodeLabel
XMLIRIValue ::= FirstArcIdentifier SubsequentArcIdentifier
RelativeIRIType ::= RELATIVE-OID-IRI
RelativeIRIValue ::= "" FirstRelativeArcIdentifier SubsequentArcIdentifier ""
FirstRelativeArcIdentifier ::= ArcIdentifier
XMLRelativeIRIValue ::= FirstRelativeArcIdentifier SubsequentArcIdentifier
EmbeddedPDVType ::= EMBEDDED PDV
EmbeddedPDVValue ::= SequenceValue
XMLEmbeddedPDVValue ::= XMLSequenceValue
ExternalType ::= EXTERNAL
ExternalValue ::= SequenceValue
XMLExternalValue ::= XMLSequenceValue
TimeType ::= TIME
TimeValue ::= tstring
XMLTimeValue ::= xmltstring

```

```

DateType ::= DATE
TimeOfDayType ::= TIME-OF-DAY
DateTimeType ::= DATE-TIME
DurationType ::= DURATION
CharacterStringType ::=
    RestrictedCharacterStringType
    | UnrestrictedCharacterStringType
CharacterStringValue ::=
    RestrictedCharacterStringValue
    | UnrestrictedCharacterStringValue
XMLCharacterStringValue ::=
    XMLRestrictedCharacterStringValue
    | XMLUnrestrictedCharacterStringValue
RestrictedCharacterStringType ::=
    BMPString
    | GeneralString
    | GraphicString
    | IA5String
    | ISO646String
    | NumericString
    | PrintableString
    | TeletexString
    | T61String
    | UniversalString
    | UTF8String
    | VideotexString
    | VisibleString
RestrictedCharacterStringValue ::=
    cstring
    | CharacterStringList
    | Quadruple
    | Tuple
CharacterStringList ::= "{" CharSyms "}"
CharSyms ::=
    CharsDefn
    | CharSyms "," CharsDefn
CharsDefn ::=
    cstring
    | Quadruple
    | Tuple
    | DefinedValue
Quadruple ::= "{" Group "," Plane "," Row "," Cell "}"
Group ::= number
Plane ::= number
Row ::= number
Cell ::= number
Tuple ::= "{" TableColumn "," TableRow "}"
TableColumn ::= number
TableRow ::= number
XMLRestrictedCharacterStringValue ::= xmlcstring
UnrestrictedCharacterStringType ::= CHARACTER STRING
UnrestrictedCharacterStringValue ::= SequenceValue

```

```

XMLUnrestrictedCharacterStringValue ::= XMLSequenceValue

UsefulType ::= typereference

ConstrainedType ::=
    Type Constraint
    | TypeWithConstraint

TypeWithConstraint ::=
    SET Constraint OF Type
    | SET SizeConstraint OF Type
    | SEQUENCE Constraint OF Type
    | SEQUENCE SizeConstraint OF Type
    | SET Constraint OF NamedType
    | SET SizeConstraint OF NamedType
    | SEQUENCE Constraint OF NamedType
    | SEQUENCE SizeConstraint OF NamedType

Constraint ::= "(" ConstraintSpec ExceptionSpec ")"

ConstraintSpec ::=
    SubtypeConstraint
    | GeneralConstraint

SubtypeConstraint ::= ElementSetSpecs

ElementSetSpecs ::=
    RootElementSetSpec
    | RootElementSetSpec ";" "... "
    | RootElementSetSpec ";" "... " ";" AdditionalElementSetSpec

RootElementSetSpec ::= ElementSetSpec

AdditionalElementSetSpec ::= ElementSetSpec

ElementSetSpec ::=
    Unions
    | ALL Exclusions

Unions ::=
    Intersections
    | UElms UnionMark Intersections

UElms ::= Unions

Intersections ::=
    IntersectionElements
    | IElems IntersectionMark IntersectionElements

IElems ::= Intersections

IntersectionElements ::= Elements | Elms Exclusions

Elms ::= Elements

Exclusions ::= EXCEPT Elements

UnionMark ::= "|" | UNION

IntersectionMark ::= "^" | INTERSECTION

Elements ::=
    SubtypeElements
    | ObjectSetElements
    | "(" ElementSetSpec ")"

SubtypeElements ::=
    SingleValue
    | ContainedSubtype
    | ValueRange
    | PermittedAlphabet
    | SizeConstraint
    | TypeConstraint

```

- | InnerTypeConstraints
- | PatternConstraint
- | PropertySettings
- | DurationRange
- | TimePointRange
- | RecurrenceRange

SingleValue ::= Value

ContainedSubtype ::= Includes Type

Includes ::= INCLUDES | empty

ValueRange ::= LowerEndpoint "." UpperEndpoint

LowerEndpoint ::= LowerEndValue | LowerEndValue "<"

UpperEndpoint ::= UpperEndValue | "<" UpperEndValue

LowerEndValue ::= Value | MIN

UpperEndValue ::= Value | MAX

SizeConstraint ::= SIZE Constraint

TypeConstraint ::= Type

PermittedAlphabet ::= FROM Constraint

InnerTypeConstraints ::=
 WITH COMPONENT SingleTypeConstraint
 | WITH COMPONENTS MultipleTypeConstraints

SingleTypeConstraint ::= Constraint

MultipleTypeConstraints ::=
 FullSpecification
 | PartialSpecification

FullSpecification ::= "{" TypeConstraints "}"

PartialSpecification ::= "{" "... " ";" TypeConstraints "}"

TypeConstraints ::=
 NamedConstraint
 | NamedConstraint ";" TypeConstraints

NamedConstraint ::= identifier ComponentConstraint

ComponentConstraint ::= ValueConstraint PresenceConstraint

ValueConstraint ::= Constraint | empty

PresenceConstraint ::= PRESENT | ABSENT | OPTIONAL | empty

PatternConstraint ::= PATTERN Value

PropertySettings ::= SETTINGS simplestring

PropertySettingsList ::=
 PropertyAndSettingPair
 | PropertySettingsList PropertyAndSettingPair

PropertyAndSettingPair ::= PropertyName "=" SettingName

PropertyName ::= psname

SettingName ::= psname

DurationRange ::= ValueRange

TimePointRange ::= ValueRange

RecurrenceRange ::= ValueRange

ExceptionSpec ::= "!" ExceptionIdentification | empty

ExceptionIdentification ::=
SignedNumber
| DefinedValue
| Type ":" Value

2.3.3.2. X.681

DefinedObjectClass ::=
ExternalObjectClassReference
| objectclassreference
| UsefulObjectClassReference

ExternalObjectClassReference ::= modulereference "." objectclassreference

UsefulObjectClassReference ::=
TYPE-IDENTIFIER
| ABSTRACT-SYNTAX

ObjectClassAssignment ::= objectclassreference "::=" ObjectClass

ObjectClass ::=
DefinedObjectClass
| ObjectClassDefn
| ParameterizedObjectClass

ObjectClassDefn ::= CLASS "{" FieldSpec "," + "}" WithSyntaxSpec?

FieldSpec ::=
TypeFieldSpec
| FixedTypeValueFieldSpec
| VariableTypeValueFieldSpec
| FixedTypeValueSetFieldSpec
| VariableTypeValueSetFieldSpec
| ObjectFieldSpec
| ObjectSetFieldSpec

PrimitiveFieldName ::=
typefieldreference
| valuefieldreference
| valuesetfieldreference
| objectfieldreference
| objectsetfieldreference

FieldName ::= PrimitiveFieldName "." +

TypeFieldSpec ::= typefieldreference TypeOptionalitySpec?

TypeOptionalitySpec ::= OPTIONAL | DEFAULT Type

FixedTypeValueFieldSpec ::=
valuefieldreference Type UNIQUE ? ValueOptionalitySpec ?

ValueOptionalitySpec ::= OPTIONAL | DEFAULT Value

VariableTypeValueFieldSpec ::= valuefieldreference FieldName ValueOptionalitySpec ?

FixedTypeValueSetFieldSpec ::= valuesetfieldreference Type ValueSetOptionalitySpec ?

ValueSetOptionalitySpec ::= OPTIONAL | DEFAULT ValueSet

VariableTypeValueSetFieldSpec ::=
valuesetfieldreference FieldName ValueSetOptionalitySpec ?

ObjectFieldSpec ::= objectfieldreference DefinedObjectClass ObjectOptionalitySpec ?

ObjectOptionalitySpec ::=
OPTIONAL

```

|   DEFAULT Object
ObjectSetFieldSpec ::=
    objectsetfieldreference DefinedObjectClass ObjectSetOptionalitySpec ?
ObjectSetOptionalitySpec ::=
    OPTIONAL
    |   DEFAULT ObjectSet
WithSyntaxSpec ::= WITH SYNTAX SyntaxList
SyntaxList ::= "{" TokenOrGroupSpec empty + "}"
TokenOrGroupSpec ::= RequiredToken | OptionalGroup
OptionalGroup ::= "[" TokenOrGroupSpec empty + "]"
RequiredToken ::= Literal | PrimitiveFieldName
Literal ::= word | ","
DefinedObject ::= ExternalObjectReference | objectreference
ExternalObjectReference ::= modulereference "." objectreference
ObjectAssignment ::= objectreference DefinedObjectClass ": :=" Object
Object ::= DefinedObject | ObjectDefn | ObjectFromObject | ParameterizedObject

```

2.3.3.3. X.682

```

GeneralConstraint ::= UserDefinedConstraint | TableConstraint | ContentsConstraint
UserDefinedConstraint ::= CONSTRAINED BY "{" UserDefinedConstraintParameter "," * "}"
UserDefinedConstraintParameter ::=
    Governor ":" Value
    |   Governor ":" Object
    |   DefinedObjectSet
    |   Type
    |   DefinedObjectClass
TableConstraint ::= SimpleTableConstraint | ComponentRelationConstraint
SimpleTableConstraint ::= ObjectSet
ComponentRelationConstraint ::= "{" DefinedObjectSet "}" "{" AtNotation "," + "}"
AtNotation ::= "@ ComponentIdList | "@. Level ComponentIdList
Level ::= "." Level | empty
ComponentIdList ::= identifier "." +

```

2.3.3.4. X.683

```

ParameterizedAssignment ::=
    ParameterizedTypeAssignment
    |   ParameterizedValueAssignment
    |   ParameterizedValueSetTypeAssignment
    |   ParameterizedObjectClassAssignment
    |   ParameterizedObjectAssignment
    |   ParameterizedObjectSetAssignment
ParameterizedTypeAssignment ::= typereference ParameterList ": :=" Type
ParameterizedValueAssignment ::= valuereference ParameterList Type ": :=" Value
ParameterizedValueSetTypeAssignment ::=
    typereference ParameterList Type ": :=" ValueSet
ParameterizedObjectClassAssignment ::=
    objectclassreference ParameterList ": :=" ObjectClass

```

```

ParameterizedObjectAssignment ::=
    objectreference ParameterList DefinedObjectClass ":" := " Object
ParameterizedObjectSetAssignment ::=
    objectsetreference ParameterList DefinedObjectClass ":" := " ObjectSet
ParameterList ::= "{" Parameter "," + "}"
Parameter ::= ParamGovernor ":" DummyReference | DummyReference
ParamGovernor ::= Governor | DummyGovernor
Governor ::= Type | DefinedObjectClass
DummyGovernor ::= DummyReference
DummyReference ::= Reference
ParameterizedReference ::=
    Reference
    | Reference "{" "}"
SimpleDefinedType ::= ExternalTypeReference | typereference
SimpleDefinedValue ::= ExternalValueReference | valuereference
ParameterizedType ::= SimpleDefinedType ActualParameterList
ParameterizedValue ::= SimpleDefinedValue ActualParameterList
ParameterizedValueSetType ::= SimpleDefinedType ActualParameterList
ParameterizedObjectClass ::= DefinedObjectClass ActualParameterList
ParameterizedObjectSet ::= DefinedObjectSet ActualParameterList
ParameterizedObject ::= DefinedObject ActualParameterList
ActualParameterList ::= "{" ActualParameter "," + "}"
ActualParameter ::= Type | Value | ValueSet | DefinedObjectClass | Object | ObjectSet

```

2.3.4. Data Registry

Data Registry는 구문 분석기를 통해 수집되는 정보를 저장하는 Block이다. 구문분석기가 객체를 Importing하는 것을 돕거나, 분석이 완료된 정보를 Intermediate File로 Exporting하는 것을 돕는다.



(그림 .5) Data Registry

- Management Tool

정보 추가 및 조회를 실행하는 API 집합이다.

- Importing Tool

Repository Path로부터 특정 객체를 Importing하는 API 집합이다.

- Exporting Tool

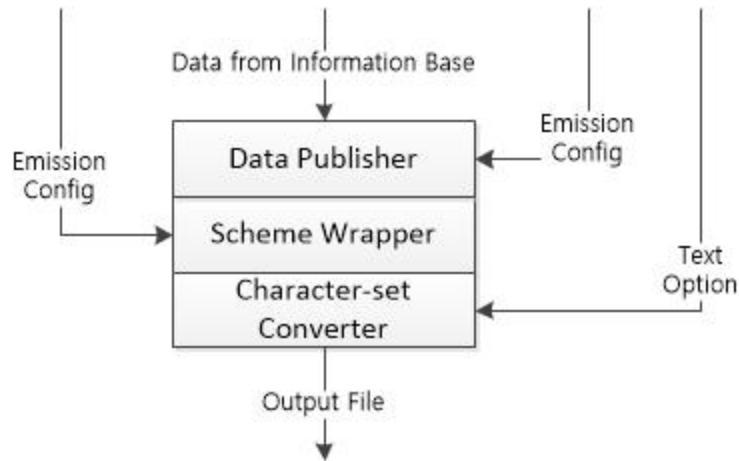
Information Base에 저장된 Target Data를 Repository Path에 저장하는 API 집합이다.

- Information Base

객체의 정보를 저장하는 자료구조이다. 빠른 검색을 하기 용이한 Indexing 자료구조를 적용하여 구현한다.

2.3.5. Emitter Block

Emitter Block은 최종 출력을 생성하는 Block이다. Information Base로부터 원본 데이터를 받아 출력값을 적절하게 변환하여, 지정 Scheme에 담아 출력한다. 출력 파일이 Text File Format인 경우 Text 출력 설정에 따라 문자셋 변환 과정을 거친다.



(그림 .6) Emitter Block

- Data Publisher

주어진 데이터의 값을 출력에 용이한 형태로 변환한다. 출력이 Text인 경우 Binary, Float Point Number, 표현할수 없는 특수문자 등의 값을 적절한 String으로 변환한다. 출력이 Binary인 경우 Endian Correction, Record Align등의 작업을 수행한다.

- Scheme Wrapper

Data Publisher에 의해 변환된 출력 Data를 주어진 Scheme에 따라 Containing한다. Text출력인 경우 JSON, XML, CSV를 지원한다.

- Character-set Converter

출력이 Text인 경우, Emission Config에 주어진 Target 문자셋으로 변환을 담당한다.

2.4. 기능 정의

2.4.1. Argument Process 기능

- o Repository 지정

 - I={path}

 - path: Intermediate File이 저장된 경로

- o Emitter Config File 지정

 - ecfg={filename}

 - filename: Emitter Config File의 파일 시스템 파일명

- o Output File 지정

 - out={filename}

 - filename: Emitter Config에 의해 설정된 출력이 저장될 파일 시스템 파일명

- o Input File 지정

 - 위에 열거하지 않은 모든 토큰을 Input File로 지정

2.4.2. Data Registry 기능

- o Add

 - Type Object 및 Value Object를 추가한다. Module Identifier, Object Name을 Key로 사용한다.

- o Find

 - Module Identifier, Object Name을 Key로 Type Object또는 Value Object를 검색한다.

- o Import Module

 - Module Identifier를 이용하여 Repository Path에 있는 File을 읽어 Information Base에 등록한다.

- o Export

 - Module Identifier에 해당하는 Module을 Intermediate Format으로 변환하여 Binary File로 저장한다.

2.4.3. Emitter 기능

- o Emission Scheme

 - 출력 파일의 Context를 특정 Scheme에 따라 출력한다. JSON, XML, CSV 및 Binary Scheme을 지원한다.

- o Emission Form

 - 출력파일에 Emit할 정보를 나열한다. 데이터 타입은 Information Base의 원시 형태를 사용하며, Emission Scheme에 따라 적절한 변환이 적용된다. Emission Form을 적용할 수 있는 Category는 다음과 같다.

 - Module Information From

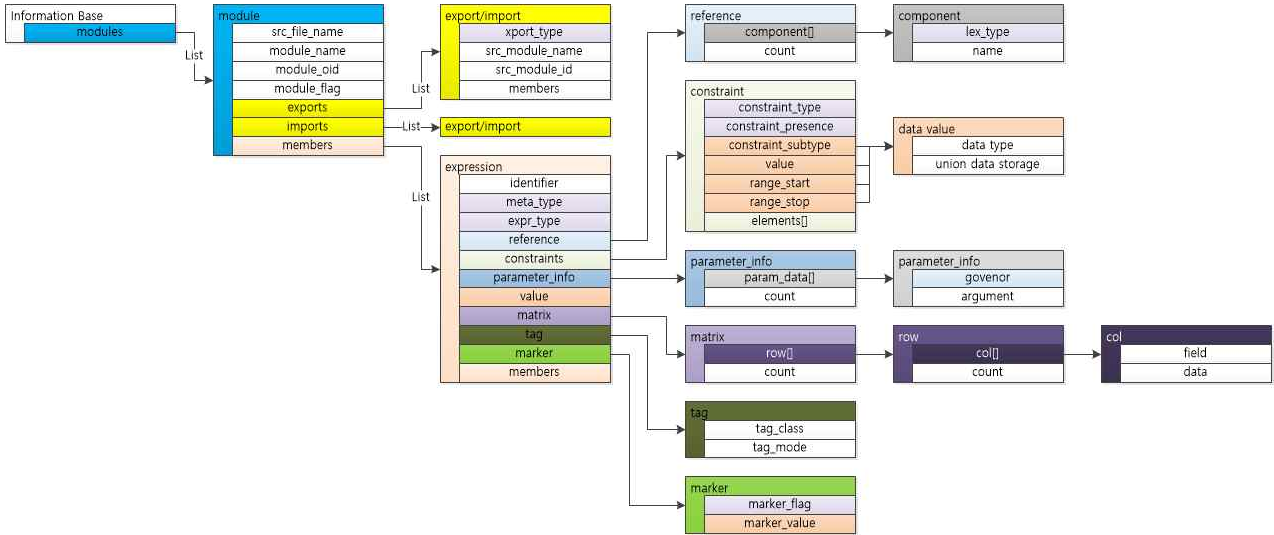
 - Type Object From

 - Value Object From

2.5. 상세 설계

2.5.1. Information Base

List link는 빠른 검색을 위해 AVL B-Tree로 구성한다.



(그림.7) Information Base

3. DB 및 DBMS 시스템 구축

3.1. 사용자 기능

일반 사용자가 이용할 수 있는 시스템의 기능에 대해 정의하며 사용자 정보 관리 기능, 사용자 로그인 관련 기능, 메인화면 및 메뉴, 공지사항 및 표준자료실, 링크 등으로 구분될 수 있다.

추가적으로 일정한 권한을 가진 사용자는 특정 메타데이터를 검색, 등록, 수정, 삭제 신청을 할 수 있는 기능이 존재한다.

(표.4) 사용자 기능 요구사항

번호	요구사항ID	구분	요구사항	요구사항 내용
1	R_U_001	사용자 정보 관리	사용자 정보 등록	사용자가 회원 가입을 통해 사용자 정보를 등록
2	R_U_002		사용자 정보 조회	사용자가 로그인 후 사용자 정보를 조회
3	R_U_003		사용자 정보 수정	사용자가 로그인 후 사용자 정보를 수정
4	R_U_004		사용자 정보 삭제	회원 가입된 사용자가 회원 탈퇴를 신청하여 사용자 정보를 삭제
5	R_U_005		사용자 정보 활용 동의	사용자로부터 정보 활용 동의 여부를 표시한 후 이를 저장
6	R_U_006		사용자 정보 활용 동의 철회	사용자로부터 정보 활용 동의 철회를 표시한 후 이를 저장
7	R_U_007	사용자 로그인	사용자 인증	사용자가 입력한 ID, 암호가 DB에 저장된 값과 일치하는지 비교 확인 이때 사용자 암호가 http를 통해 직접적으로 전송되지 않아야 함
8	R_U_008		사용자 로그인	사용자가 ID, 암호를 입력하여 서버에 로그인
9	R_U_009		사용자 로그아웃	사용자가 로그아웃하여 웹브라우저에 사용자 정보 삭제
10	R_U_010		사용자 자동 로그아웃	사용자가 일정 기간 사이트를 이용하지 않을 경우 Session timeout에 의한 자동 로그아웃됨
11	R_U_011		사용자 로그인 차단	사용자가 동일한 계정으로 동시 로그인할 경우 두번째 사용자 로그인을 차단
12	R_U_012		사용자 계정 잠금	사용자가 암호 입력을 5회 연속 오류 입력 시 해당 계정 잠금
13	R_U_013		사용자 본인 확인	ID, 암호를 잊은 사용자가 email, SMS 등 2차 수단으로 본인 정보를 조회
14	R_U_014		사용자 암호 초기화	사용자 암호를 초기화 함
15	R_U_015		사용자 계정 잠금 해제	사용자 계정 잠금 기능으로 로그인이 불가능한 사용자의 계정 잠금을 해제
16	R_U_016		사용자 접속 정보 등록	사용자가 접속한 ip, 웹브라우저 타입 등을 저장
17	R_U_017	사용자 메인화면	메인화면 조회	사용자가 Header, Main, Footer 등으로 구성된 메인페이지를 조회
18	R_U_018	사용자 메뉴	메뉴 목록 조회	사용자가 메뉴 목록을 조회
19	R_U_019		메뉴 조회	사용자가 메뉴 선택 후 하부 메뉴 조회 및 선택 가능
20	R_U_020	사용자 공지사항	공지사항 목록 조회	사용자가 공지사항 목록을 조회
21	R_U_021		공지사항 조회	사용자가 공지사항의 내용을 확인
22	R_U_022	사용자 표준자료실	표준자료 목록 조회	사용자가 표준자료 목록을 조회
23	R_U_023		표준자료 조회	사용자가 표준자료 내용을 조회
24	R_U_024	사용자 링크	링크 목록 조회	사용자가 링크목록을 조회
25	R_U_025	사용자 메타데이터 연동	검색입력에 의한 표준정보 조회	사용자가 검색조건을 입력한 후 이에 해당하는 표준 정보를 조회
26	R_U_026		검색된 표준 정보와 첨부 파일 다운로드 기능	검색된 메타데이터 정보에서 표준 정보등의 정보를 txt, excel, pdf 파일로 다운로드
27	R_U_027		메타데이터 등록 신청	사용자가 메타데이터를 등록
28	R_U_028		메타데이터 수정 신청	사용자가 메타데이터 수정을 요청
29	R_U_029		메타데이터 삭제 신청	사용자가 메타데이터 삭제를 요청
30	R_U_030		메타데이터 신청내역 이력 조회	특정 사용자가 신청한 메타데이터 신청 내역을 조회

3.2. 관리자 일반 기능

관리자가 이용할 수 있는 시스템의 관리자 일반 기능에 대해 정의하며 사용자 정보 관리, 사용자 그룹 권한 관리, 메인화면 관리, 메뉴 관리, 공지사항 관리, 표준자료실 관리, 링크 관리 등으로 구분될 수 있다.

(표.5) 관리자 일반 기능 요구사항

번호	요구사항ID	구분	요구사항	요구사항 내용
1	R_A_001	사용자 정보 관리	사용자 가입 목록 조회	관리자가 사용자 가입 목록을 조회
2	R_A_002		사용자 가입 내역 조회	관리자가 가입된 특정 사용자 정보를 조회
3	R_A_003		사용자 가입 승인	사용자의 가입 요청을 승인
4	R_A_004		사용자 암호 초기화	가입된 사용자의 암호를 초기화
5	R_A_005		사용자 그룹 목록 조회	사용자가 소속된 그룹의 목록을 조회
6	R_A_006		사용자 그룹 조회	사용자가 소속된 그룹의 정보를 조회
7	R_A_007		사용자 그룹 추가	사용자를 특정 그룹에 추가
8	R_A_008		사용자 그룹 수정	사용자가 소속된 그룹의 정보를 수정
9	R_A_009		사용자 그룹 삭제	특정 사용자 그룹을 삭제
10	R_A_010		사용자 권한 목록 조회	사용자에 부여할 수 있는 권한을 조회
11	R_A_011		사용자 권한 조회	특정 사용자의 권한을 조회
12	R_A_012		사용자 권한 추가	특정 사용자에게 권한을 부여
13	R_A_013		사용자 권한 수정	특정 사용자의 권한을 수정
14	R_A_014		사용자 권한 삭제	특정 사용자의 권한을 삭제
15	R_A_015	사용자 그룹 권한 관리	사용자 그룹 권한 목록 조회	사용자 그룹에게 부여할 수 있는 권한을 조회
16	R_A_016		사용자 그룹 권한 조회	특정 사용자 그룹의 권한을 조회
17	R_A_017		사용자 그룹 권한 추가	특정 사용자 그룹에 권한을 추가
18	R_A_018		사용자 그룹 권한 수정	특정 사용자 그룹에 권한을 수정
19	R_A_019		사용자 그룹 권한 삭제	특정 사용자 그룹의 권한을 삭제
20	R_A_020	메인화면 관리	메인화면 이미지 조회	초기 메인화면의 메인 이미지를 조회
21	R_A_021		메인화면 이미지 변경	초기 메인화면의 메인 이미지를 변경
22	R_A_022		메인화면 구성 조회	메인화면의 구성을 조회
23	R_A_023		메인화면 구성 변경	메인화면의 구성을 변경
24	R_A_024		배너 이미지 조회	메인화면의 배너 이미지를 조회
25	R_A_025		배너 이미지 변경	메인화면의 배너 이미지를 변경
26	R_A_026	메뉴 관리	메뉴 목록 조회	메인화면의 메뉴 목록을 조회
27	R_A_027		메뉴 조회	메인화면의 특정 메뉴를 조회
28	R_A_028		메뉴 등록	메인화면에 새로운 메뉴를 등록
29	R_A_029		메뉴 수정	메인화면의 특정 메뉴를 수정
30	R_A_030		메뉴 삭제	메인화면의 특정 메뉴를 삭제
31	R_A_031		메뉴 접근권한 설정	메인화면의 특정 메뉴의 권한을 설정
32	R_A_032		메뉴 사용여부 설정	메인화면의 특정 메뉴의 사용여부를 설정
33	R_A_033	공지 사항 관리	공지사항 목록 조회	공지사항 목록을 조회
34	R_A_034		공지사항 조회	특정 공지사항 내용을 조회
35	R_A_035		공지사항 등록	새로운 공지사항을 등록
36	R_A_036		공지사항 수정	특정 공지사항 내용을 수정
37	R_A_037		공지사항 삭제	특정 공지사항을 삭제
38	R_A_038	표준 자료실 관리	표준자료 목록 조회	표준자료 목록을 조회
39	R_A_039		표준자료 조회	특정 표준자료의 내용을 조회
40	R_A_040		표준자료 등록	새로운 표준자료를 등록
41	R_A_041		표준자료 수정	특정 표준자료의 내용을 수정
42	R_A_042		표준자료 삭제	특정 표준자료를 삭제
43	R_A_043	링크 관리	링크 목록 조회	링크 목록을 조회
44	R_A_044		링크 조회	특정 링크 내용을 조회
45	R_A_045		링크 등록	새로운 링크 내용을 등록
46	R_A_046		링크 수정	특정 링크 내용을 수정
47	R_A_047		링크 삭제	특정 링크를 삭제

3.3. 관리자 메타데이터 연동 기능

관리자가 이용할 수 있는 메타데이터 연동 기능에 대해 정의하며, 이는 메타데이터 관리, 사용자 신청 메타데이터 관리로 구분될 수 있다.

(표.6) 관리자 메타데이터 연동 기능 요구사항

번호	요청사항ID	구분	요구사항	요구사항 내용
1	R_M_001	메타데이터 관리	자동 추출 메타데이터 목록 조회	자동 추출된 메타데이터의 목록을 조회
2	R_M_002		자동 추출 메타데이터 조회	특정 자동 추출 메타데이터의 내용을 조회
3	R_M_003		자동 추출 메타데이터 등록	새로운 자동 추출 메타데이터를 등록
4	R_M_004		자동 추출 메타데이터 수정	특정 자동 추출 메타데이터의 내용을 수정
5	R_M_005		자동 추출 메타데이터 삭제	특정 자동 추출 메타데이터를 삭제
6	R_M_006		자동 추출 메타데이터 변경이력 조회	특정 자동 추출 메타데이터의 변경이력을 조회
7	R_M_007		수동 메타데이터 목록 조회	수동 입력으로 등록된 메타데이터 목록을 조회
8	R_M_008		수동 메타데이터 조회	특정 수동 입력 메타데이터 내용을 조회
9	R_M_009		수동 메타데이터 등록	새로운 수동 입력 메타데이터를 등록
10	R_M_010		수동 메타데이터 수정	특정 수동 입력 메타데이터의 내용을 수정
11	R_M_011		수동 메타데이터 삭제	특정 수동 입력 메타데이터를 삭제
12	R_M_012		수동 메타데이터 변경이력 조회	특정 수동 입력 메타데이터의 변경이력을 조회
13	R_M_013	사용자 신청 메타데이터 관리	메타데이터 등록/수정/삭제 신청 목록 조회	사용자가 신청한 메타데이터의 처리 요청 목록을 조회
14	R_M_014		메타데이터 등록/수정/삭제 신청 내역 조회	특정 사용자가 신청한 메타데이터의 처리 요청 내역을 조회
15	R_M_015		메타데이터 등록/수정/삭제 신청 승인	특정 사용자가 신청한 메타데이터의 처리 요청을 승인
16	R_M_016		메타데이터 등록/수정/삭제 신청 거절	특정 사용자가 신청한 메타데이터의 처리 요청을 거절
17	R_M_017		메타데이터 등록/수정/삭제 신청 거절 사유 등록	특정 사용자가 신청한 메타데이터의 처리 요청을 거절하고 이에 대한 사유를 등록

3.4. 관리자 시스템 연동 기능

관리자가 이용할 수 있는 전용 컴파일러 연동 기능에 대해 정의한다.

(표.7) 관리자 시스템 연동 기능 요구사항

번호	요청사항ID	구분	요구사항	요구사항 내용
1	R_I_001	컴파일러 연동	표준문서(ASN.1) 로드 및 각종 파라미터 입력	사용자의 파라미터,파일등의 입력을 받을 수 있는 Web 화면을 준비
2	R_I_002		전용 컴파일러 실행 및 메타데이터 추출	사용자로부터 전달받은 입력을 이용하여 전용 컴파일러를 실행 및 메타데이터를 추출
3	R_I_003		전용 컴파일러 실행 및 메타데이터 추출 결과 이력 등록	전용 컴파일러를 실행한 결과와 메타데이터 추출 오류를 이력 목록에 등록
4	R_I_004		전용 컴파일러 실행 및 메타데이터 추출 오류 이력 등록	전용 컴파일러를 실행한 결과와 메타데이터 추출 결과를 이력 목록에 등록
5	R_I_005		전용 컴파일러 실행 결과 이력 조회	전용 컴파일러 실행 결과에 대한 이력을 조회
6	R_I_006		전용 컴파일러 실행 오류 이력 조회	전용 컴파일러 실행 오류에 대한 이력을 조회
7	R_I_007		전용 컴파일러 동작 테스트	전용 컴파일러를 가상으로 실행하고 정상적으로 실행되는지 테스트

3.5. 관리자 시스템 운영 통계 기능

관리자가 이용할 수 있는 시스템 운영 상태와 메타데이터 수집 통계 기능에 대해 정의한다.

(표.8) 관리자 시스템 운영 통계 기능 요구사항

번호	요청사항ID	구분	요구사항	요구사항 내용
1	R_S_001	시스템 운영 상태/사용 통계	메타데이터 수집 통계	수집된 메타데이터에 대한 기간별 통계 조회
2	R_S_002		메타데이터 조회 통계	수집된 메타데이터에 대한 기간별 검색 통계 조회
3	R_S_003		메타데이터 등록 오류 내역 조회	메타데이터 등록 시 발생된 오류 내역 조회
4	R_S_004		원천데이터 요청 접수/승인 전송 통계	원천 데이터 요청 접수/승인 비율에 대한 통계 정보 조회
5	R_S_005		메뉴별 이용자 수 현황	사용자가 이용한 메뉴 중 Top 5 조회
6	R_S_006		데이터 검색 수 이용 현황	사용자가 이용한 검색 중 Top 10 통계 조회

3.6. S/W 설계

메타데이터 관리시스템의 내부 구조를 보다 간략하게 표현하기 위해서 시스템 내부 SW 블록과 각 SW 블록의 연동 방식을 기술한다.

3.6.1. NginX(Web Server)

NginX는 웹서버 소프트웨어로, 가벼움과 높은 성능을 목표로 하며 웹서버, 리버스 프록시(Reverse Proxy) 등의 기능을 가지고 있다. 웹서버는 일반적으로 미리 만들어진 html 파일(정적파일)을 다수의 접속자에게 동시에 서비스할 수 있도록 설계되어 있다.

3.6.2. Gunicorn(WSGI Server)

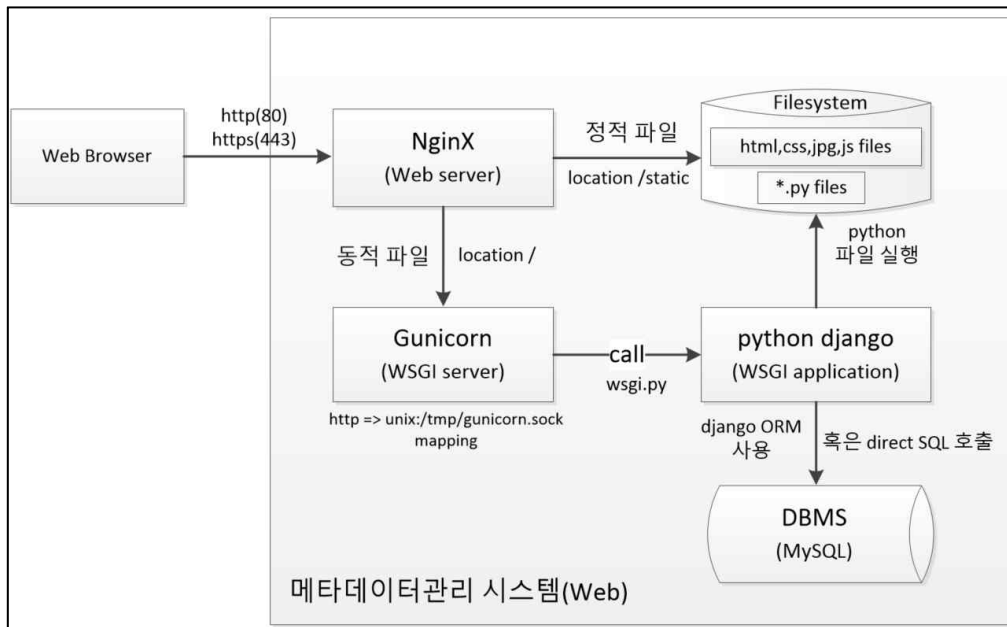
특정 언어로 된 코드를 웹서버와 연동하기 위해서는 서로 통신할 수 있는 게이트웨이(gateway) 역할을 하는 매개체가 필요한데 그 다리 역할을 하는 것이 WSGI(Web Server Gateway Interface) Server 이다. Gunicorn은 웹서버와 python을 연동해 주는 소프트웨어이다.

3.6.3. Python Django(WSGI Application)

Python Django는 python으로 만들어진 무료 오픈 소스(open source) 웹 애플리케이션 프레임워크(web application framework)이다. 쉽고 빠르게 웹사이트를 개발할 수 있도록 돕는 구성요소로 이루어진 웹 프레임워크이다. 메타데이터 관리시스템은 Python Django 웹 프레임워크를 이용하여 작성된 웹 애플리케이션이다.

3.6.4. RDBMS(MySQL)

관계형 데이터베이스를 관리하는 시스템으로 테이블 행(컬럼) 단위로 검색, 삽입, 갱신, 삭제를 쉽게 해주는 관리 소프트웨어이다



(그림 .8) 메타데이터 관리시스템의 SW 블록 구조

위의 그림은 사용자의 특정 요청이 메타데이터 관리시스템 내부에서 처리되는 절차를 나타내는 논리적인 SW 블록 구성도이다.

3.7. H/W(클라우드) 설계

메타데이터 관리시스템은 다양한 클라우드 환경에서 적용이 가능하고 서비스에 대한 확장성이 용이하도록 가상화 기술을 이용하여 H/W(Virtual H/W) 블록을 설계한다.

3.7.1. VM(Virtual Machine) 기반 가상화 기술

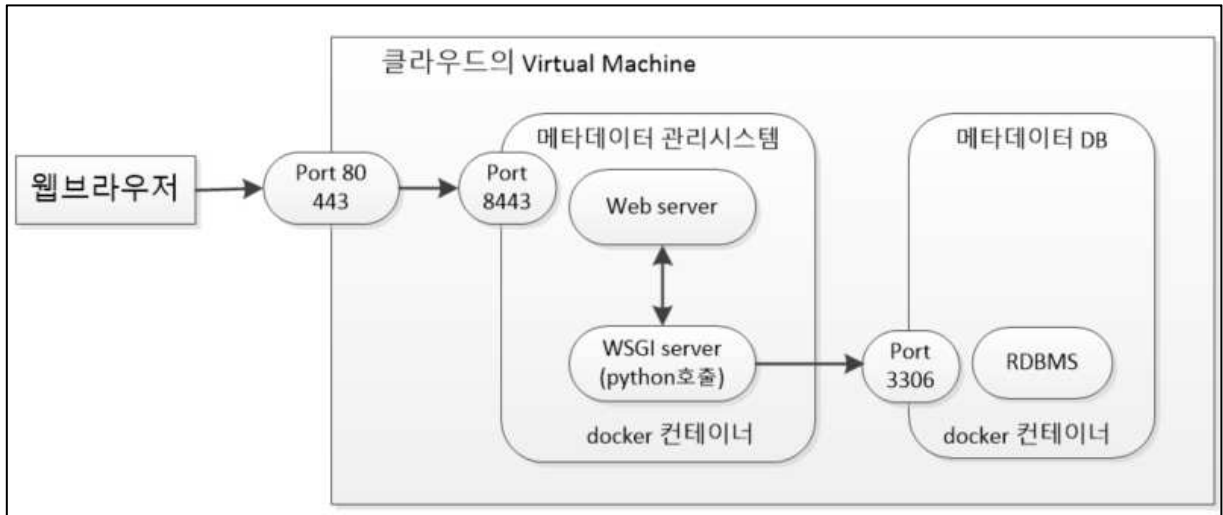
단일 물리 서버의 CPU에서 여러 VM을 실행하며 VM 간 애플리케이션을 격리하고 애플리케이션의 정보를 다른 애플리케이션에서 자유롭게 접근할 수 없도록 일정 수준의 보안성을 제공한다. 직접적인 H/W 장비 사용 혹은 베어메탈(Bare-Metal) 시스템 대비 시스템 자원을 보다 효율적으로 활용할 수 있으며, 쉽게 애플리케이션을 추가하거나 업데이트할 수 있고 하드웨어 비용을 절감할 수 있어 더 나은 확장성을 제공한다.

3.7.2. Container 기반 가상화 기술

Container(컨테이너) 기반 가상화 기술은 VM 기반 가상화 기술과 유사하지만 격리 속성을 완화하여 애플리케이션 간에 운영체제(OS)를 공유하므로 각각의 컨테이너는 일반 VM 보다 CPU와 시스템 자원을 덜 사용하므로 성능면에서 더 나은 성능을 제공한다. 컨테이너 기반 가상화 기술에는 일반적으로 Docker Container 기술이 사용된다.

3.7.3. 가상화 기반 메타데이터 관리시스템 블록 구성도

아래 그림은 클라우드 서비스의 VM(Virtual Machine)에 컨테이너 기반 가상화 기술을 적용한 메타데이터관리 시스템 Docker 컨테이너와 메타데이터 DB Docker 컨테이너가 설치 및 상호 운용되는 모습을 도식화한 그림이다.



(그림.9) 컨테이너 가상화 기반 메타데이터 관리시스템의 블록 구성도

클라우드 서비스 외부에 존재하는 사용자는 웹 브라우저를 통해서 클라우드 서비스의 VM에 사용자 요청(http request)을 80/443 port를 사용하여 전달하고 VM은 해당 요청을 다시 8443 port를 사용하여 메타데이터관리 시스템 컨테이너에 전달한다.

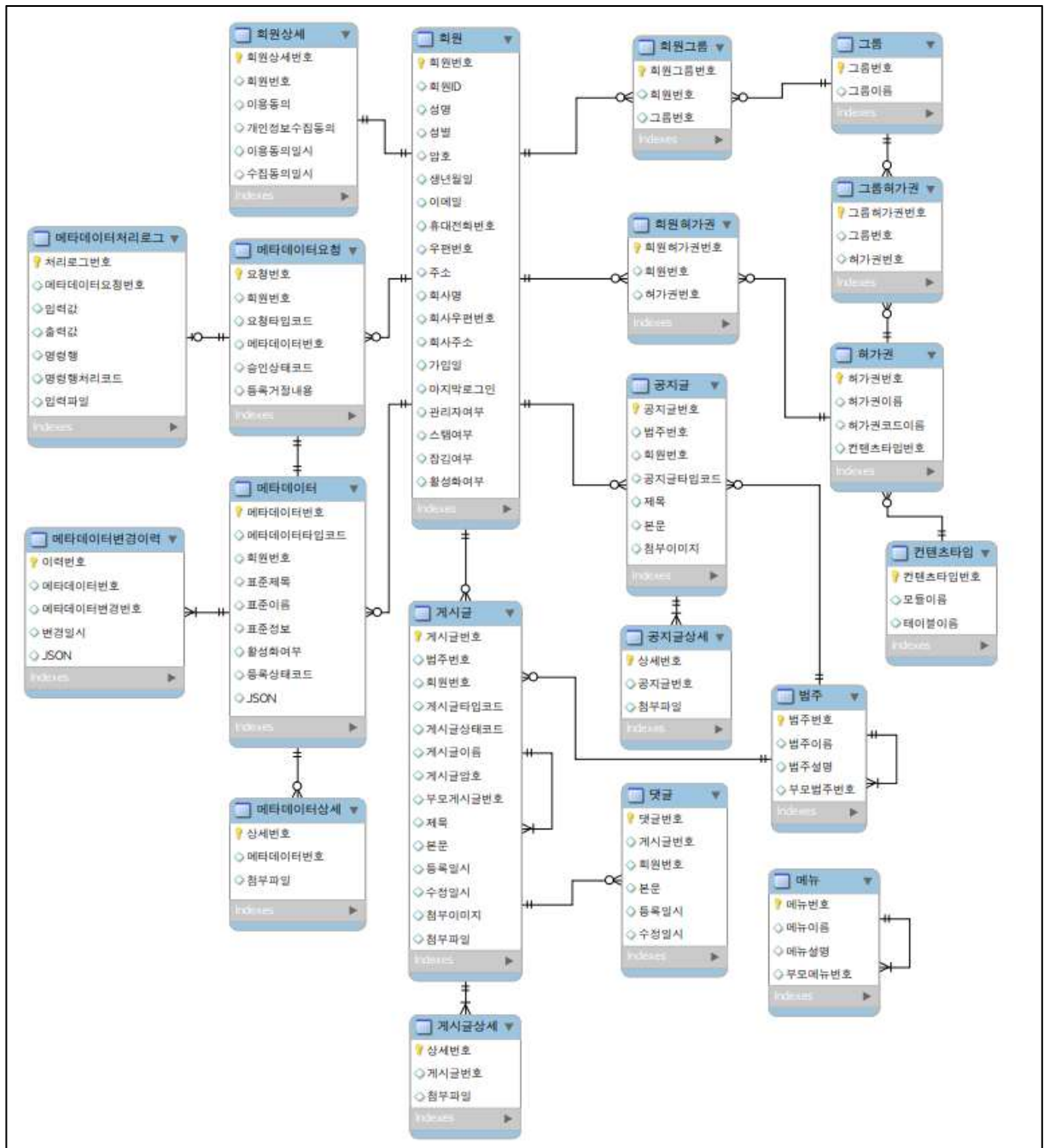
메타데이터 관리시스템 컨테이너에서 전달받은 요청(http request)은 해당 컨테이너에 존재하는 웹 서버에게 전달되고, 웹 서버는 다시 해당 요청을 WSGI(Web Server Gateway Interface) Server를 통해서 python 소스와 연결하여 해당 요청을 처리한다.

WSGI Server가 호출한 python 소스에는 메타데이터 DB를 구동하고 있는 메타데이터 컨테이너의 IP, port(3306)와 Database Name, user ID, user password 등의 정보가 들어있으며 이러한 정보를 이용하여 메타데이터 관리시스템 컨테이너에서 메타데이터 DB 컨테이너로 접속할 수 있다.

3.8. DB 설계

3.8.1. 논리 모델(Logical Model) 설계

논리 모델 설계에서는 엔티티와 엔티티 타입 및 관계를 정의한다. 즉 어떠한 정보를 객체화할 것인가에 설계한다. 아래 ERD는 메타데이터 관리시스템의 논리 모델 ERD를 도식화한 그림이다.

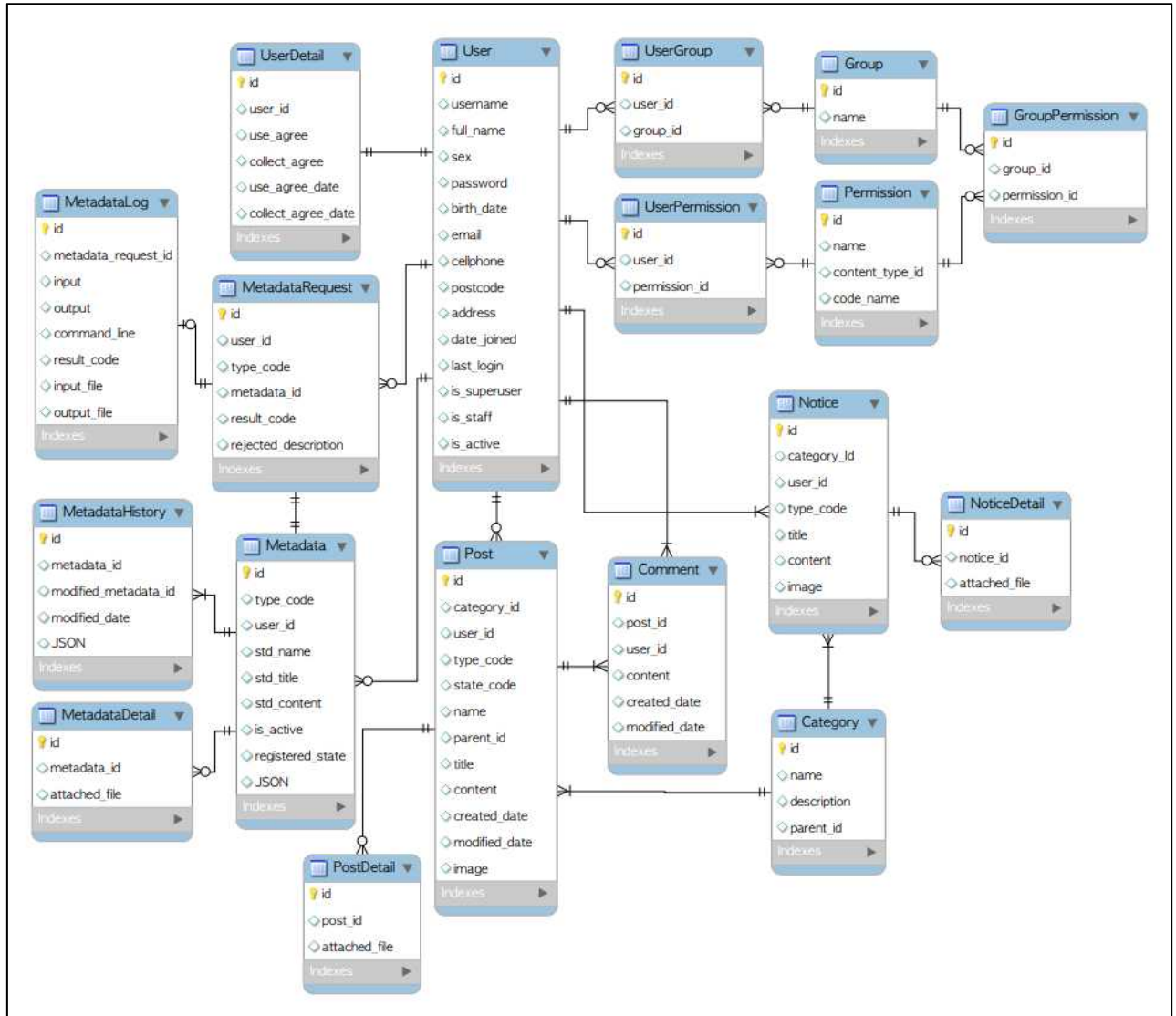


(그림 .10) 메타데이터 관리시스템의 논리적 ERD

3.8.2. 물리 모델(Physical Model) 설계

물리 모델 설계에서는 각 엔티티의 관계에 따라 나올 수 있는 물리 테이블 즉 실제 DBMS에 생성될 테이블을 설계한다. 이 외에도 관계에 대한 정의, 인덱스, 컬럼 별 데이터 타입 및 제약조건 등의 속성을 구체적으로 정의하며 정규화를 실시한다.

아래 ERD는 메타데이터 관리시스템의 물리 모델 ERD를 도식화한 그림이다.



(그림 .11) 메타데이터 관리시스템의 물리적 ERD

3.8.3. 테이블 정의

- 사용자와 그룹 관련 테이블

3.8.3.1. User

사용자의 정보 중 기본 정보에 해당 내용을 저장하는 테이블이다.

(표.9) User 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
auth_user (User)	사용자 기본 정보	id	int	primary key	O	N
		password	varchar(128)	사용자 암호		N
		last_login	datetime(6)	마지막 로그인한 시간		
		is_superuser	tinyint(1)	관리자 여부		N
		username	varchar(150)	사용자 ID	O	N
		first_name	varchar(150)	사용자 이름		N
		last_name	varchar(150)	사용자 성		N
		email	varchar(254)	사용자 email		N
		is_staff	tinyint(1)	운영자 여부		N
		is_active	tinyint(1)	사용자 활성화 여부		N
date_joined	datetime(6)	사용자 가입 날짜		N		

3.8.3.2. UserDetails

사용자의 정보 중 부가 정보에 해당 내용을 저장하는 테이블이다.

(표.10) User Detail 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
common_user_profile (UserProfile)	사용자 부가 정보	id	int	primary key	O	N
		cellphone	longtext	사용자 휴대전화 번호		N
		address	longtext	사용자 주소		N
		sex_code	varchar(1)	사용자 성별		N
		birth_date	datetime(6)	사용자 생년월일		
		user_id	int	User테이블 id의 foreign key	O	N

3.8.3.3. Group

사용자를 특정 목적에 따라 구분할 목적으로 사용하는 그룹에 대한 정보를 저장하는 테이블이다.

(표.11) Group 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
auth_group (Group)	그룹 정보	id	int	primary key	O	N
		name	varchar(150)	그룹 이름	O	N

3.8.3.4. UserGroup

사용자가 속하는 그룹에 대한 정보를 저장하는 테이블이다.

(표.12) User Group 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
auth_user_groups (UserGroup)	사용자 그룹 정보	id	int	primary key	O	N
		user_id	int	User테이블 id의 foreign key	O	N
		group_id	int	Group테이블 id의 foreign key	O	N

○ 허가권 관련 테이블

3.8.3.5. Permission

사용자나 사용자가 속한 그룹에 대한 권한 정보를 저장하는 테이블이다.

(표.13) Permission 테이블

테이블명 (ORM 모델명)	테이블 정보	필드명	데이터타입	필드설명	인덱스	NULL
auth_permissions (Permission)	허가권 정보	id	int	primary key	O	N
		name	varchar(255)	permission 이름	O	N
		content_type_id	int	ContentType테이블 id의 foreign key	O	N
		codename	varchar(100)	permission code	O	N

3.8.3.6. UserPermission

특정 사용자의 권한 정보를 저장하는 테이블이다.

(표.14) User Permission 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
auth_user_user_permissions (UserPermission)	사용자별 허가권 정보	id	int	primary key	O	N
		user_id	int	User테이블 id의 foreign key	O	N
		permission_id	int	Permission테이블 id의 foreign key	O	N

3.8.3.7. GroupPermission

특정 그룹의 권한 정보를 저장하는 테이블이다.

(표.15) Group Permission 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
auth_group_permissions (GroupPermission)	그룹별 허가권 정보	id	int	primary key	O	N
		group_id	int	Group테이블 id의 foreign key	O	N
		permission_id	int	Permission테이블 id의 foreign key	O	N

3.8.3.8. ContentType

권한 설정이 필요한 패키지과 모델(테이블)에 대한 정보를 저장하는 테이블이다.

(표.16) Content Type 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
django_content_type (ContentType)	허가권 설정을 위한 콘텐츠 타입 정보	id	int	primary key	O	N
		app_label	varchar(100)	패키지 이름	O	N
		model	varchar(100)	모델 이름	O	N

○ 공지글과 게시판글 관련 테이블

3.8.3.9. Category

공지글이나 일반글의 분류 정보를 저장하는 테이블이다.

(표.17) Category 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
common_category (Category)	글 Category 정보	id	int	primary key	O	N
		name	varchar(50)	분류 이름	O	N
		description	varchar(128)	분류 내용		
		parent_id	int	부모의 id	O	N

3.8.3.10. Notice

공지글의 기본 정보를 저장하는 테이블이다.

(표.18) Notice 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
common_notice (Notice)	공지글 정보	id	int	primary key	O	N
		subject	varchar(512)	공지글 제목	O	N
		content	logntext	공지글 내용		N
		type_code	varchar(2)	공지글 타입 코드		N
		create_date	datetime(6)	공지글 생성 시간		N
		modify_date	datetime(6)	공지글 수정 시간		N
		category_id	int	Category테이블 id의 foreign key	O	N
		user_id	int	User테이블 id의 foreign key	O	N

3.8.3.11. NoticeDetail

공지글의 부가 정보를 저장하는 테이블이다.

(표.19) NoticeDetail 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
common_notice_detail (NoticeDetail)	공지글 부가 정보	id	int	primary key	O	N
		detail_key	varchar(256)	부가 정보 key		N
		detail_value	varchar(512)	부가 정보 value		N
		attach_file	varchar(100)	첨부 파일		
		notice_id	int	User테이블 id의 foreign key	O	N

3.8.3.12. Post

일반 게시글의 기본 정보를 저장하는 테이블이다.

(표.20) Post 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
common_post (Post)	게시글 정보	id	int	primary key	O	N
		name	varchar(512)	게시글 종류	O	N
		subject	varchar(512)	게시글 제목	O	N
		content	logntext	게시글 내용		N
		type_code	varchar(2)	게시글 타입 코드		N
		state_code	varchar(2)	게시글 게시 상태		N
		create_date	datetime(6)	게시글 생성 시간		N
		modify_date	datetime(6)	게시글 수정 시간		
		category_id	int	Category테이블 id의 foreign key	O	N
		parent_id	int	부모 게시글 id의 foreign key	O	N
user_id	int	User테이블 id의 foreign key	O	N		

3.8.3.13. PostDetail

일반 게시글의 부가 정보를 저장하는 테이블이다.

(표.21) Post Detail 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
common_post_detail (PostDetail)	게시글 부가 정보	id	int	primary key	O	N
		detail_key	varchar(256)	부가 정보 key		N
		detail_value	varchar(512)	부가 정보 value		N
		attach_file	varchar(100)	첨부 파일		
		post_id	int	Post테이블 id의 foreign key	O	N

3.8.3.14. Comment

일반 게시글의 댓글 정보를 저장하는 테이블이다.

(표.22) Common 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
common_comment (Comment)	댓글 정보	id	int	primary key	O	N
		content	varchar(512)	댓글 내용		N
		create_date	datetime(6)	댓글 생성 일시		N
		modify_date	datetime(6)	댓글 수정 일시		
		post_id	int	Post테이블 id의 foreign key	O	N
		user_id	int	User테이블 id의 foreign key	O	N

○ Metadata 관련 테이블

3.8.3.15. Metadata

Metadata의 기본 정보를 저장하는 테이블이다.

(표.23) Metadata 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
metadata_metadata (Metadata)	메타데이터 기본 정보	id	int	primary key	O	N
		type_code	varchar(2)	metadata 타입 코드		N
		std_name	varchar(128)	표준이름	O	N
		std_subject	varchar(512)	표준제목	O	N
		std_content	varchar(2048)	표준내용		N
		is_active	tinyint(1)	활성화 여부		N
		register_date	datetime(6)	등록 일시		N
		json	json	metadata 실제 payload(json)		
		user_id	int	User테이블 id의 foreign key	O	N

3.8.3.16. MetadataDetail

Metadata의 부가 정보를 저장하는 테이블이다.

(표.24) MetadataDetail 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
metadata_metadata_detail (MetadataDetail)	메타데이터 부가 정보	id	int	primary key	O	N
		detail_key	varchar(256)	부가 정보 key		N
		detail_value	varchar(512)	부가 정보 value		
		attach_file	varchar(100)	첨부 파일		
		metadata_id	int	Metadata테이블 id의 foreign key	O	N

3.8.3.17. MetadataRequest

Metadata 처리 요청(자동 및 수동) 정보를 저장하는 테이블이다.

(표.25) MetadataRequest 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
metadata_metadata_request (MetadataRequest)	메타데이터 요청 정보	id	int	primary key	O	N
		type_code	varchar(2)	요청 타입 코드		N
		result_code	varchar(2)	처리 결과 코드		N
		result_description	varchar(2048)	처리 결과 내용		N
		metadata_id	int	Metadata테이블 id의 foreign key	O	N
		user_id	int	User테이블 id의 foreign key	O	N

3.8.3.18. MetadataLog

전용컴파일러 연동 시 전용컴파일러 입력값, 출력값, command line 호출 정보, 전용컴파일러 리턴값 등을 저장하는 테이블이다.

(표.26) MetadataLog 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
metadata_metadata_log (MetadataLog)	메타데이터 로그 정보	id	int	primary key	O	N
		process_code	varchar(255)	처리 내용		N
		input	longtext	입력(STDIN) 파라미터 내용		N
		output	longtext	출력(STDOUT) 내용		N
		command_line	longtext	command line 호출 내용		N
		input_file	varchar(100)	입력 파일		
		output_file	varchar(100)	출력 파일		
		request_id	int	MetadataRequest테이블 id의 foreign key	O	N

3.8.3.19. MetadataHistory

특정 Metadata의 정보가 변경(삭제 포함)되었을 해당 이력 정보를 저장하는 테이블이다.

(표.27) MetadataHistory 테이블

테이블명 (ORM 모델명)	테이블 용도	필드명	데이터타입	필드설명	인덱스	NULL
metadata_metadata_history (MetadataHistory)	메타데이터 이력 정보	id	int	primary key	O	N
		modify_date	datetime(6)	변경 이력 발생 일시		
		process_type	varchar(255)	처리 내용		N
		json	json	변경 후 metadata 결과값(json)		
		metadata_id	int	Metadata테이블 id의 foreign key	O	N
		modify_metadata_id	int	수정된 Metadata테이블 id의 foreign key	O	N

4. UI 설계

4.1. 사용자 페이지

WEB에서 사용자가 사용하게 될 페이지의 기능을 구현한 User Interface이다.

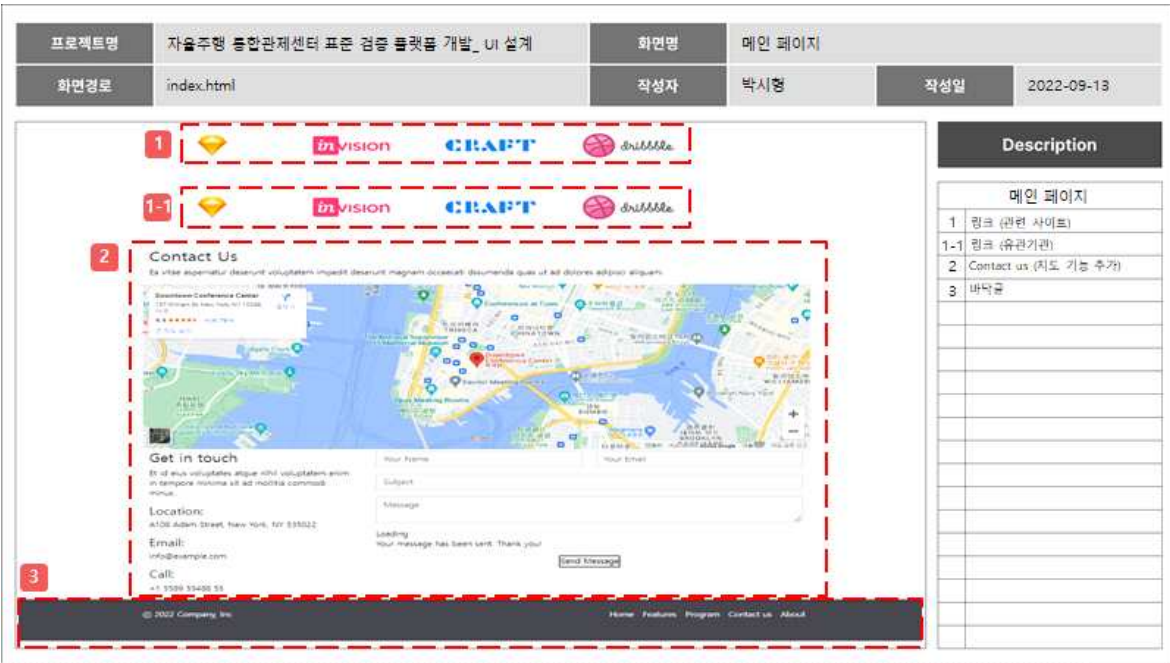
4.1.1. 메인화면

메인화면의 Header와 Navigate bar를 로그인 유/무에 따라 설계되었으며, 공지사항과 표준자료의 대표글 및 유관기관과 관련 사이트 링크 그리고 바닥글로 구성된다.

프로젝트명	자율주행 종합관제센터 표준 검증 플랫폼 개발 UI 설계	화면명	메인 페이지
화면경로	index.html	작성자	박시현
		작성일	2022-09-13

Description	
메인 페이지	
1	헤더 (관리시 해당 페이지 이동)
2	네비바 (관리시 해당 페이지 이동)
2-1	네비바 (검색 기능)
3	메너 이미지 (자동 슬라이드)
4	게시판 형태 (관리시 페이지 이동)

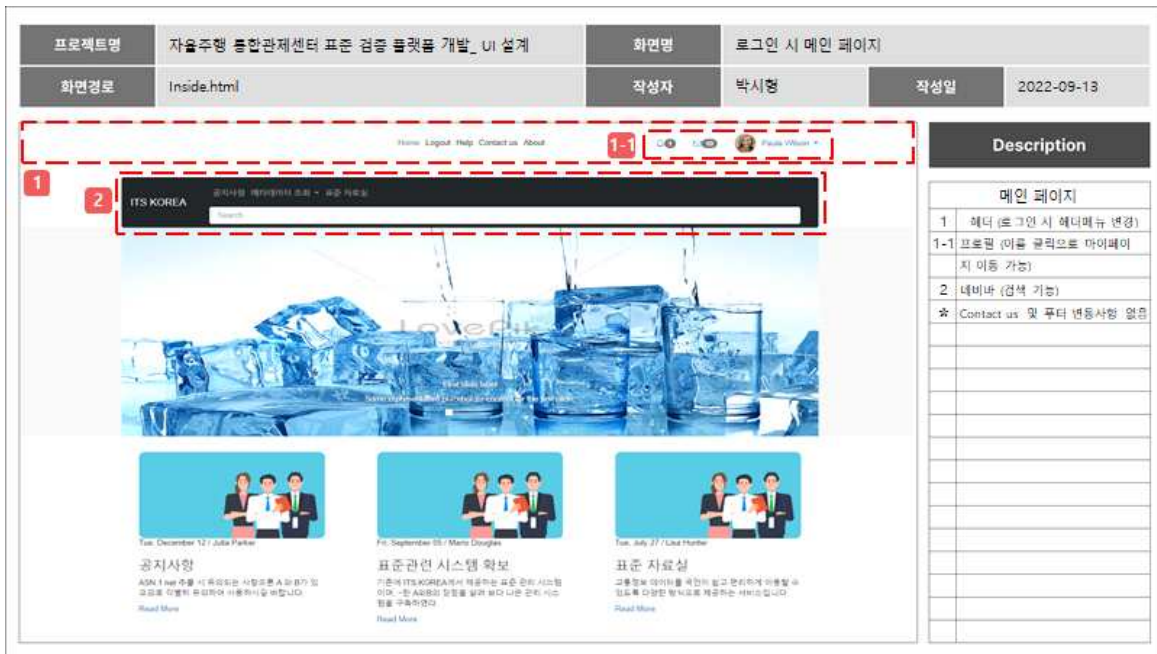
(그림.12) 사용자 페이지 - 메인화면 상단



(그림.13) 사용자 페이지 - 메인화면 하단

4.1.2. 로그인 시 메인화면

Header에 로그인 시에 표시될 프로필이 추가되었으며, 클릭 시 마이페이지로 이동할 수 있다.



(그림.14) 사용자 페이지 - 로그인 시 메인화면

4.1.3. 로그인

사용자가 로그인을 시도했을 때 디스플레이 될 User Interface이다. 비밀번호를 찾는 기능과 회원가입으로 갈 수 있는 Register 링크 및 로그인 버튼으로 구성되어 있다.

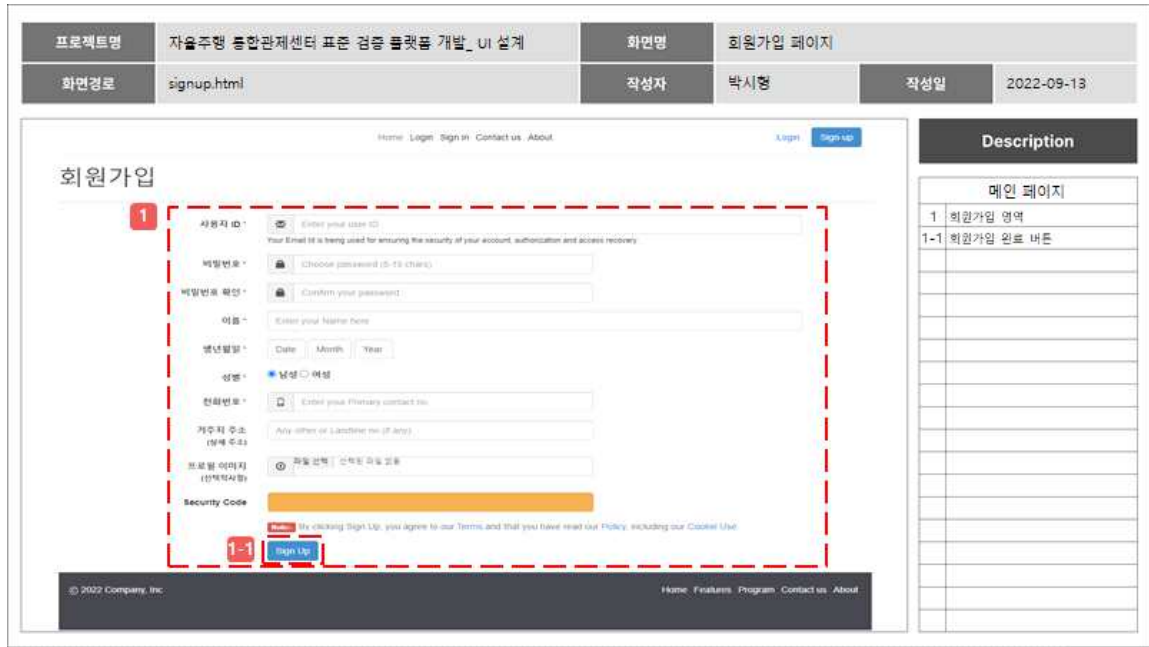
프로젝트명	자율주행 종합관리센터 포준 검증 플랫폼 개발_ UI 설계	화면명	로그인 페이지		
화면경로	login.html	작성자	박시형	작성일	2022-09-13

Description	
메인 페이지	
1	로그인 영역
1-1	비밀번호 찾기
1-2	로그인 기능 버튼
1-3	회원가입 페이지로 이동

(그림 .15) 사용자 페이지 - 로그인 화면

4.1.4. 회원가입

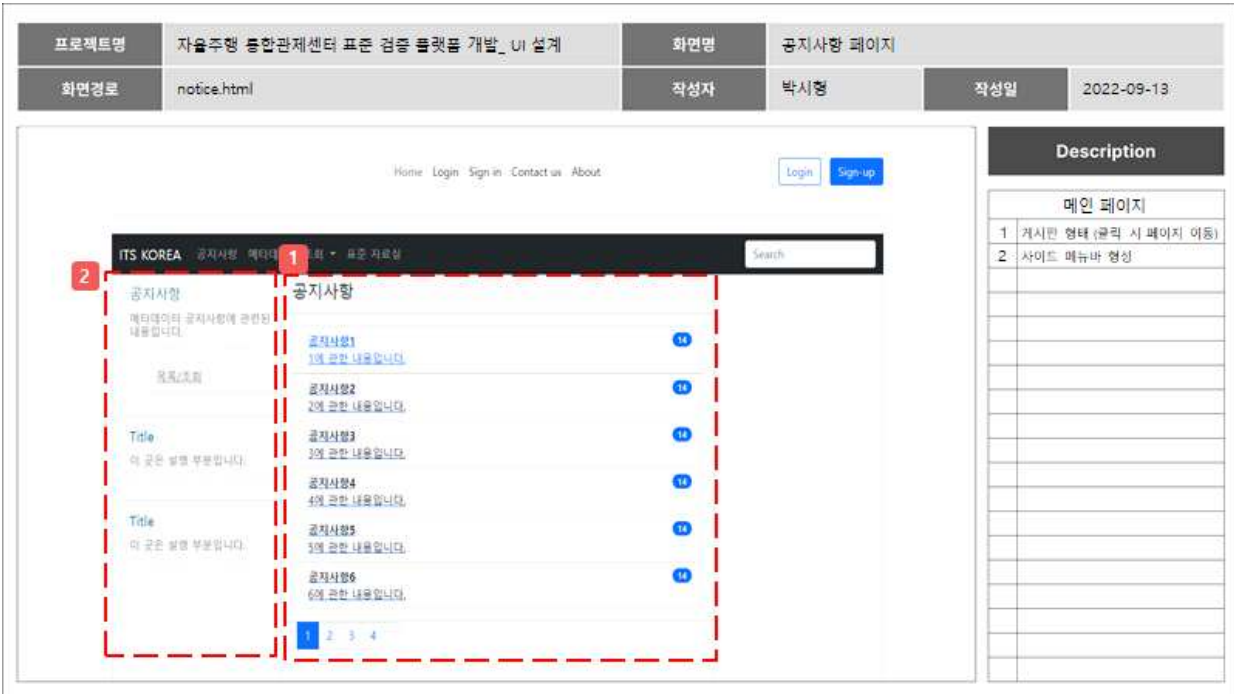
가입에 절차가 다를 수 있으므로 임의로 만들어진 회원가입의 화면이다. 가장 기본이 되는 사용자 ID, 비밀번호, 이름, 생년월일, 성별, 주소 기입란으로 형성되어 있다.



(그림.16) 사용자 페이지 - 회원가입 화면

4.1.5. 공지사항

메인화면의 첫 번째 메뉴를 맡은 카테고리이며, 사용자 페이지에서 사용자는 업로드 되는 게시글들을 읽기만 할 수 있다. 리스트들이 나열돼 있으며, 클릭 시 게시글을 볼 수 있다.



(그림.17) 사용자 페이지 - 공지사항 화면



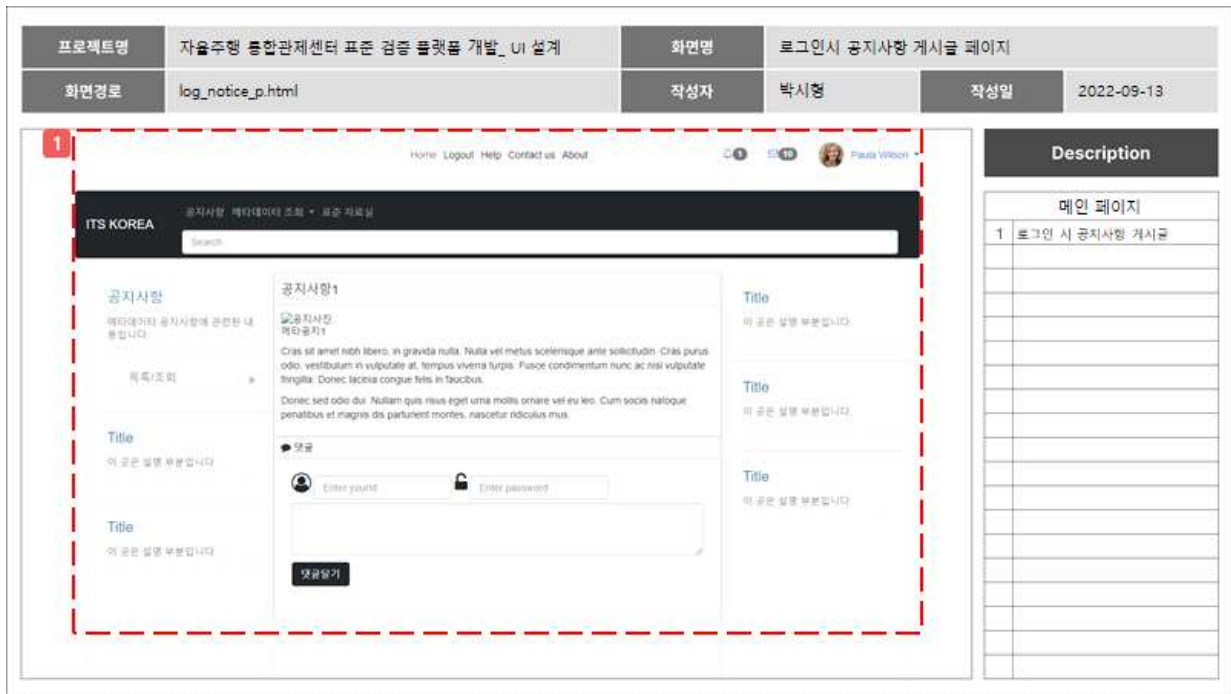
(그림.18) 사용자 페이지 - 공지사항 게시글 화면

4.1.6. 로그인 시 공지사항

로그인 시 Header에 프로필이 추가된 상태의 공지사항 User Interface이다.



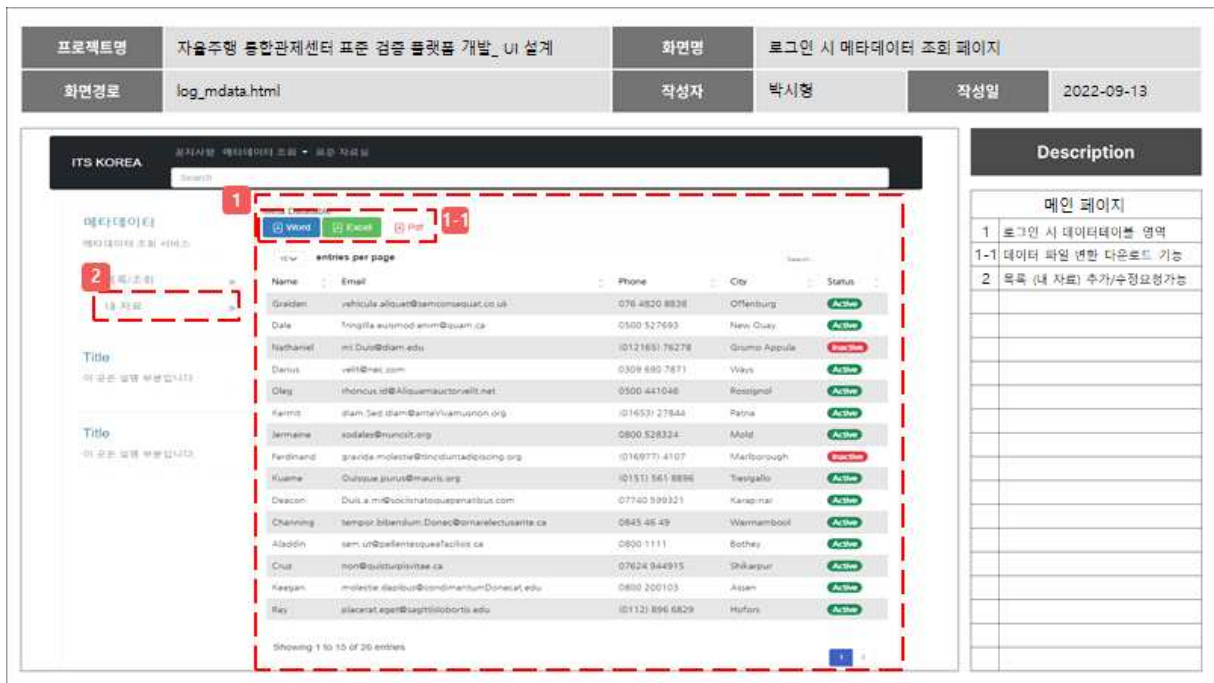
(그림.19) 사용자 페이지 - 로그인 시 공지사항 화면



(그림.20) 사용자 페이지 - 로그인 시 공지사항 게시글 화면

4.1.6.1. Meta data 조회

추출된 메타데이터는 DB와 연동되어 WEB에서 다룰 수 있게 표시될 것이며, 임의로 구성된 조회 User Interface이다. (txt, 엑셀, pdf 등)다운로드 할 수 있는 버튼들도 구성되어 있다.



(그림.21) 사용자 페이지 - 로그인 시 메타데이터 조회

4.1.7. 로그인 시 메타데이터 수정/등록 영역

사용자는 로그인 시 메타데이터 조회와 동시에 메타데이터에 관련하여 수정, 등록 신청을 할 수 있다. 수정 진행 여부를 '내 자료'에서 확인 가능한 User Interface이다.



(그림 .22) 사용자 페이지 - 로그인 시 메타데이터 수정/등록 신청 화면

4.1.7.1. 메표준 자료실

공지사항과 마찬가지로 표준 자료의 목록을 조회 가능하며, 해당 목록 클릭 시 표준 자료의 내용을 볼 수 있다. 사용자는 읽는 것 외에 등록 혹은 수정할 수 없다.

프로젝트명	자율주행 종합관제센터 표준 검증 플랫폼 개발_ UI 설계	화면명	표준 자료실 페이지		
화면경로	notice3.html	작성자	박시정	작성일	2022-09-13

Home Login Sign in Contact us About [Login](#) [Sign-up](#)

ITS KOREA 공지사항 메타데이터 조회 표준 자료실 Search

표준자료실

표준자료실입니다.

[목록/조회](#)

Title

이 곳은 설명 부분입니다.

Title

이 곳은 설명 부분입니다.

표준자료 목록

표준자료들의 최신 뉴스를 잡하세요

1-1

표준 자료 1 3 days ago

표준 자료 1에 대한 내용입니다.

2022.09.01

표준 자료 2 3 days ago

표준 자료 2에 대한 내용입니다.

2022.09.01

표준 자료 3 3 days ago

표준 자료 3에 대한 내용입니다.

2022.09.01

Title

이 곳은 설명 부분입니다.

Title

이 곳은 설명 부분입니다.

Title

이 곳은 설명 부분입니다.

Description

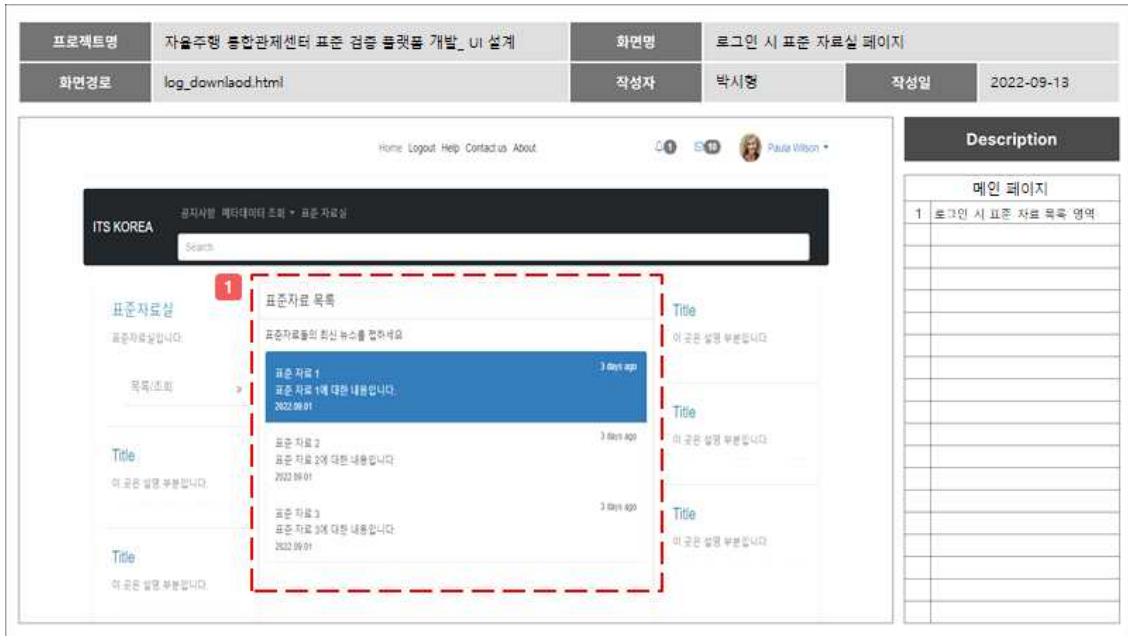
메인 페이지

1	표준 자료 목록 영역
1-1	클릭 시 표준자료 게시물 이동

(그림.23) 사용자 페이지 - 표준 자료실 화면

4.1.8. 로그인 시 표준 자료실 화면

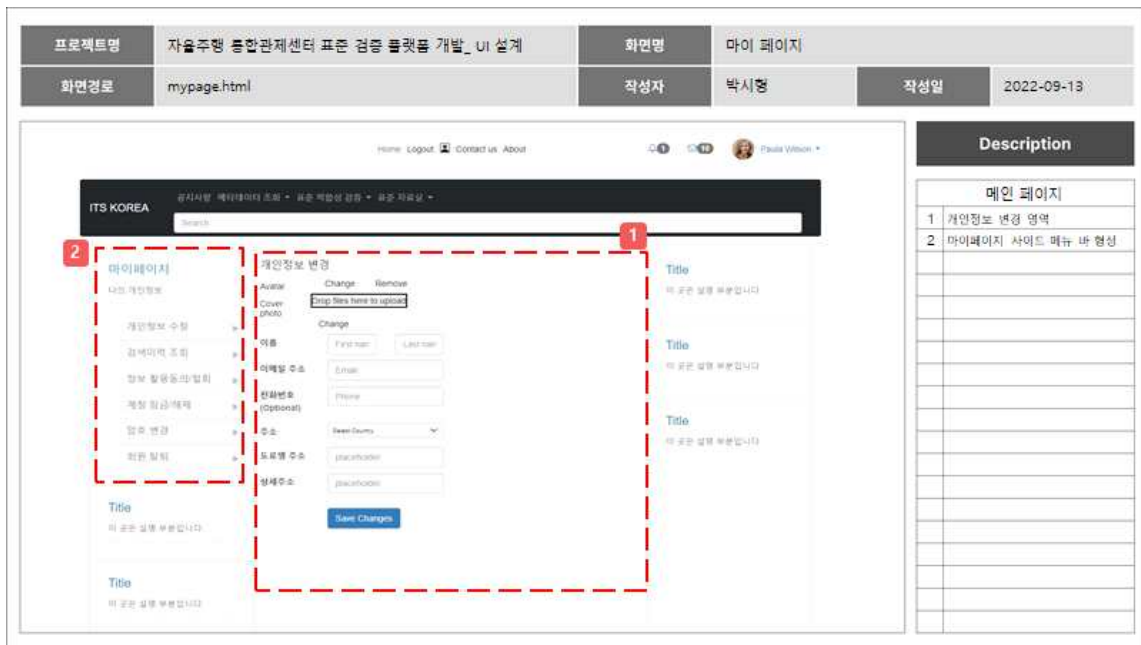
로그인 시 프로필이 추가된 Header와 같이 표시될 표준 자료실의 User Interface이다.



(그림.24) 사용자 페이지 - 로그인 시 표준 자료실 화면

4.1.8.1. My Page

로그인 시 Header에 있는 프로필 클릭으로 접근 가능하며, 사용자의 개인정보 관리를 담당하는 User Interface이다. 첫 화면으로는 개인정보를 변경할 수 있는 화면으로 구성되어 있다.



(그림.25) 사용자 페이지 - 개인정보 변경 화면

4.1.9. 관리자 페이지

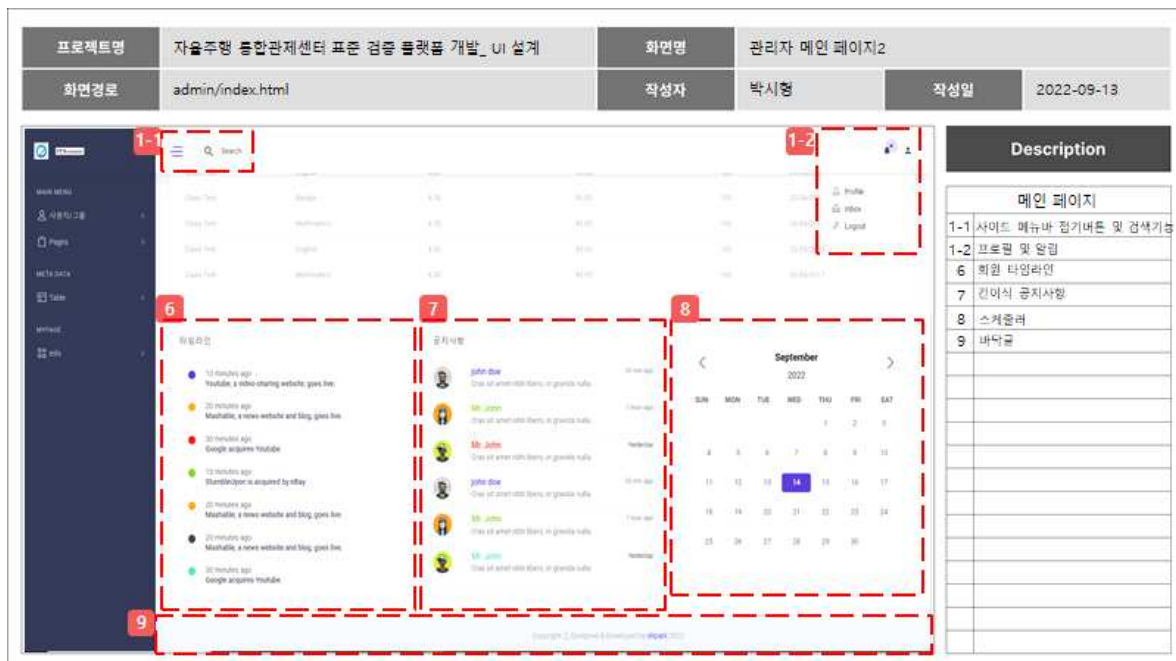
관리자가 사용자의 정보, 인증, 권한 등을 관리할 수 있으며, 화면에 표시될 메뉴, 이미지, 공지사항, 자료 등의 모든 것을 등록, 수정, 삭제할 수 있다.

4.1.9.1. 메인화면

관리자 페이지의 메인인 되는 화면이며, Dashboard에는 간이식으로 중요 기능들을 축소해서 볼 수 있다. 축소될 기능에는 메타데이터 조회, 공지사항, 일정표 등이 있으며 관리자 페이지의 메뉴들은 좌측에 표시되었다. Header도 사용자 페이지와 마찬가지로 로그인 시 포함될 프로필과 좌측에 메뉴들을 접을 수 있는 기능도 포함되어 있다.



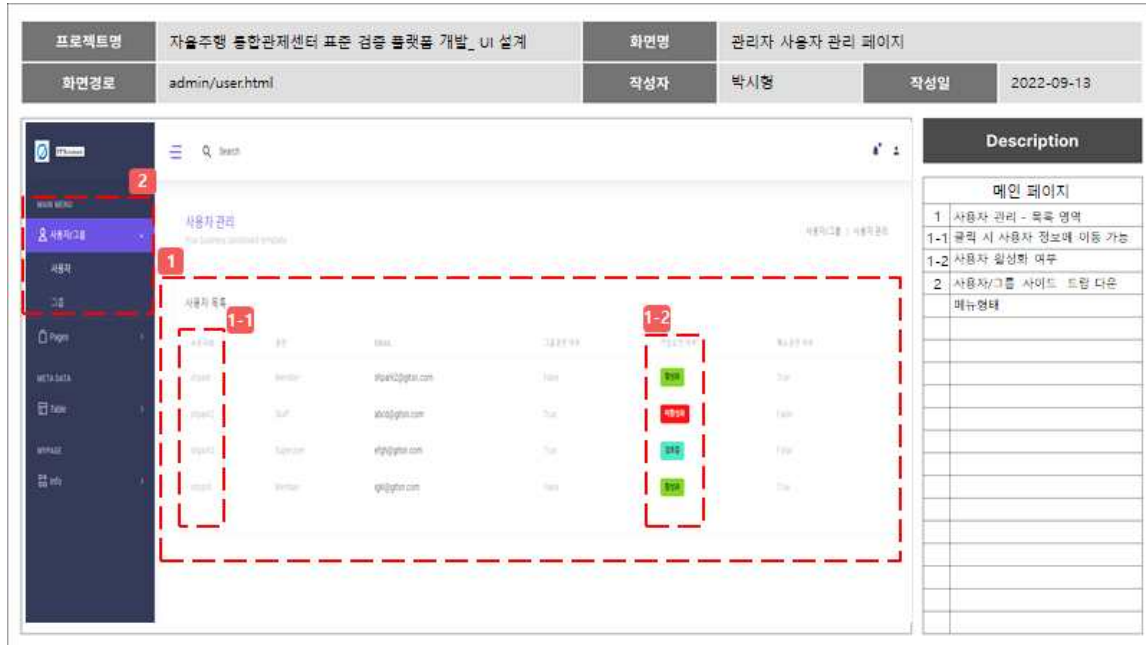
(그림 .30) 관리자 페이지 - 메인화면 상단



(그림 .31) 관리자 페이지 - 메인화면 하단

4.1.9.3. 사용자 관리

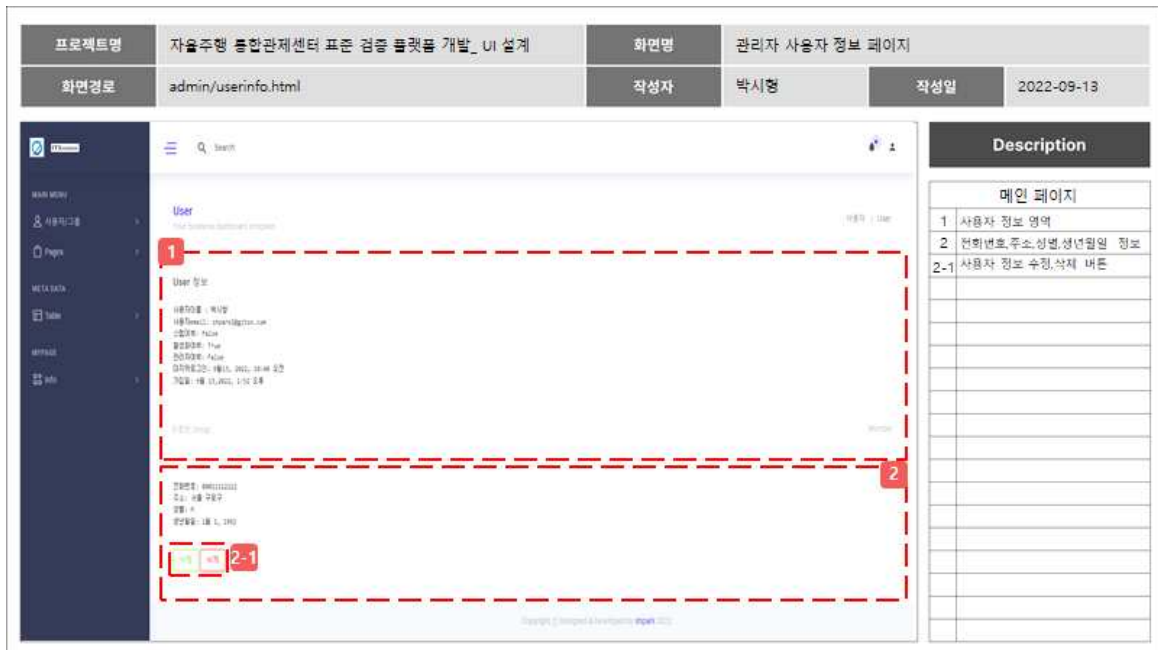
사용자들을 Table 형태에서 리스트업할 수 있으며, 사용자의 ID, 권한 상태, 그룹 및 메뉴 권한 여부, 가입 승인 여부를 확인할 수 있다.



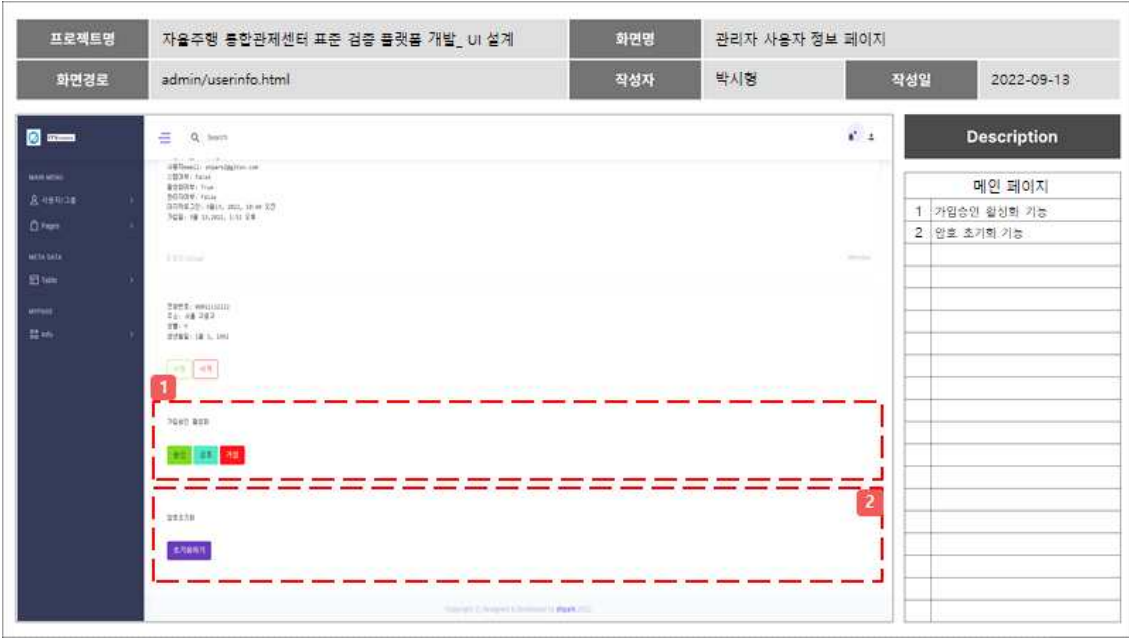
(그림.34) 관리자 페이지 - 사용자 관리

a. 사용자 정보 관리

사용자 목록에 있는 사용자 클릭 시 사용자의 정보를 수정, 삭제할 수 있으며, 가입 승인 활성화 및 암호 초기화도 가능한 기능들이 있는 User Interface이다.



(그림.35) 관리자 페이지 - 사용자 정보 화면 상단



(그림.36) 관리자 페이지 - 사용자 정보 화면 하단

b. 사용자 정보 수정

사용자 정보 화면에서 수정 기능을 눌렀을 때 표시될 화면이며, 이름, 이메일 주소, 그룹 및 그룹과 메뉴 권한 등을 설정할 수 있는 영역의 User Interface이다.



(그림.37) 관리자 페이지 - 사용자 정보 수정 화면

c. 사용자 암호 초기화

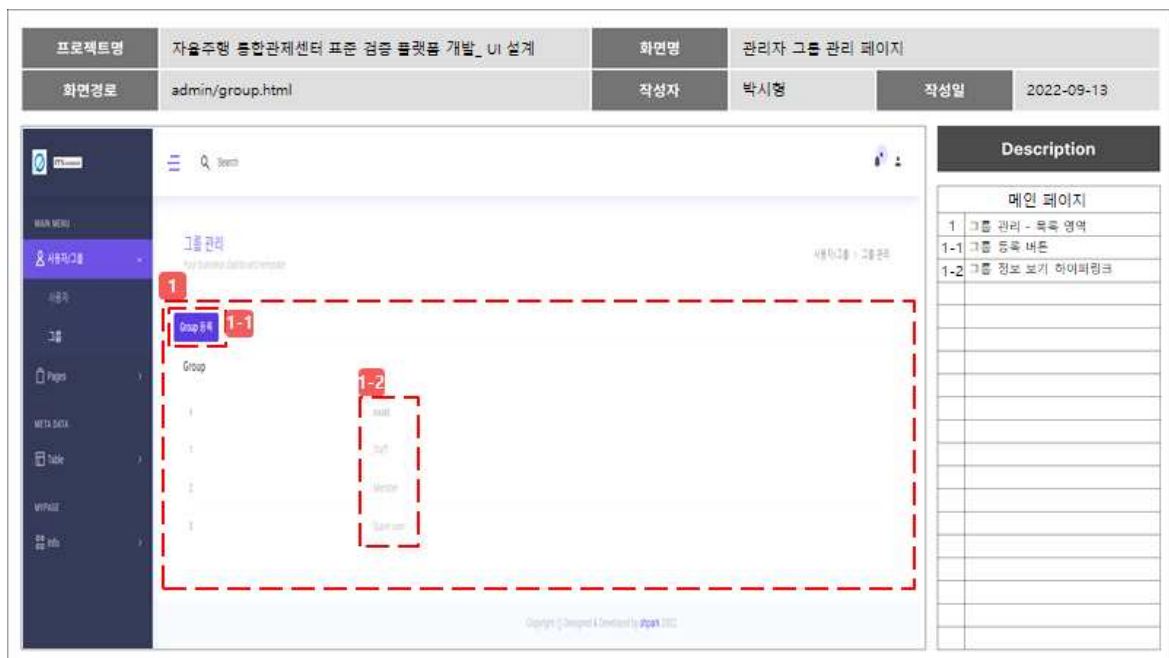
사용자 정보 화면에서 암호 초기화 클릭 시 표시될 화면이며, 암호 변경이 가능하다.



(그림.38) 관리자 페이지 - 사용자 암호 초기화 화면

4.1.9.4. 그룹 관리

관리자가 그룹의 종류별로 나열되어있는 Table로 확인 가능하며, 상단에 있는 그룹 등록 버튼을 통해 그룹들을 추가할 수 있다.



(그림.39) 관리자 페이지 - 그룹 관리 화면

a. 그룹 등록

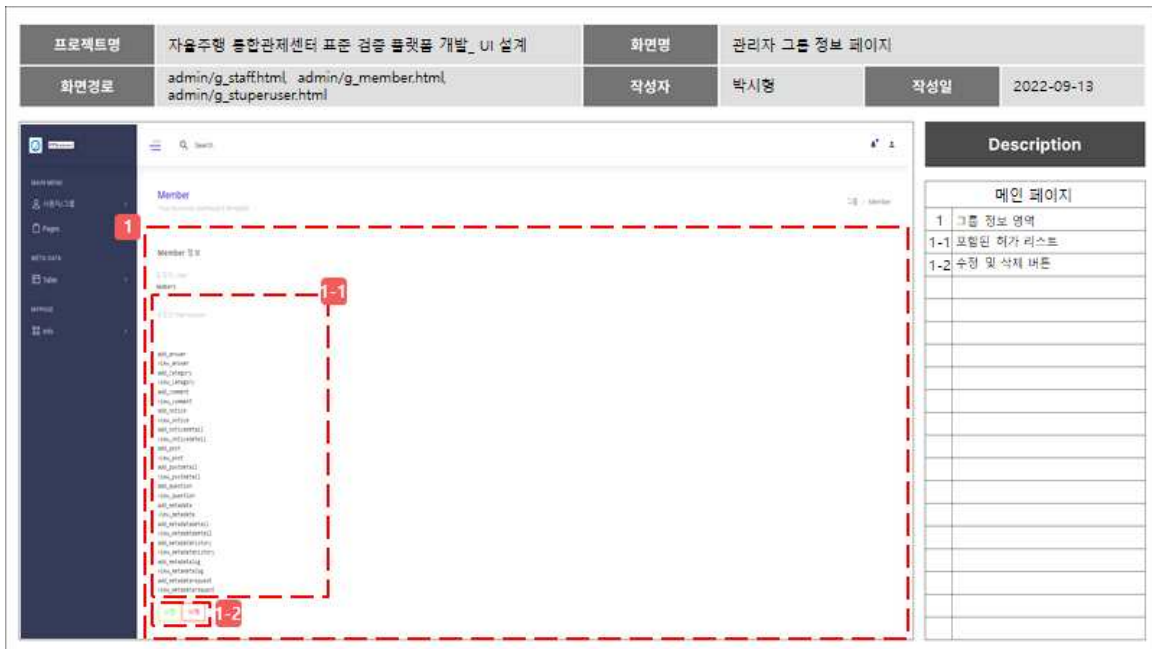
그룹 이름을 표기한 뒤, 등록 버튼을 통해 Multi box list에 표기할 그룹을 추가할 수 있다.



(그림.40) 관리자 페이지 - 그룹 등록 화면

b. 그룹 정보

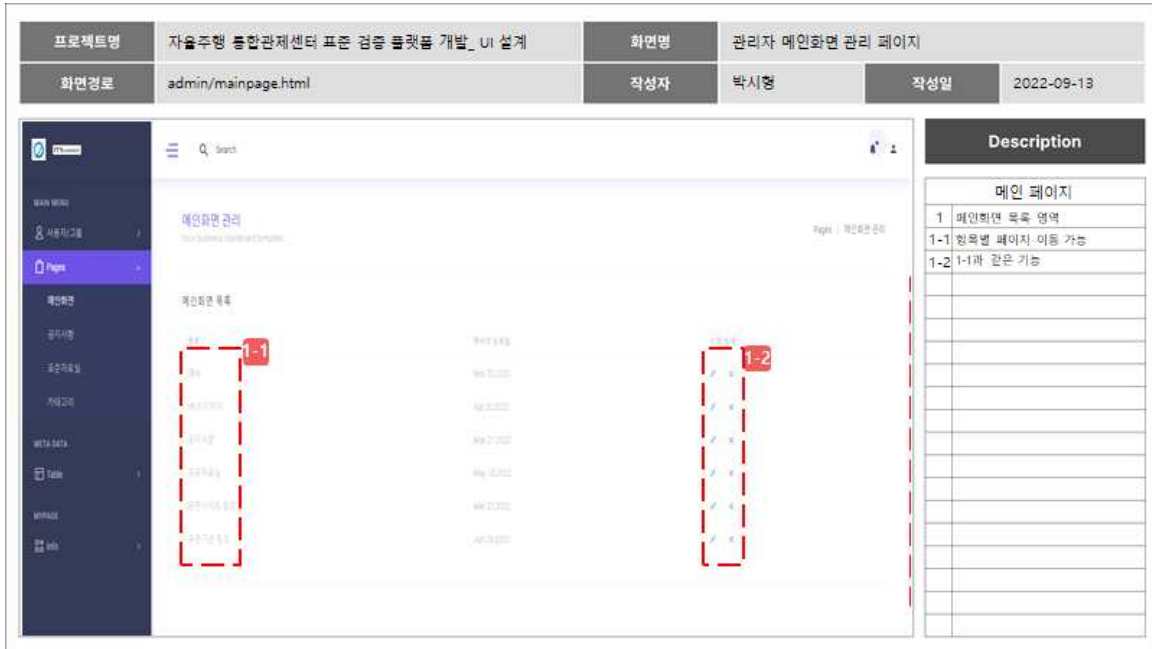
그룹에 포함된 권한을 확인할 수 있으며, 수정 기능을 통해 그룹이 가진 권한들을 추가할 수 있다.



(그림.41) 관리자 페이지 - 그룹 정보 화면

4.1.9.5. 메인화면 관리

메인화면에 디스플레이 될 메뉴, 배너 이미지, 공지사항, 표준 자료실, 관련 사이트 링크와 유관기관 링크를 등록 및 수정할 수 있다. 각 항목들을 클릭 시 수정할 수 있는 페이지로 이동한다.



(그림.42) 관리자 페이지 - 메인화면 관리

a. 메뉴 관리

사용자 페이지에서 보일 메뉴를 추가, 수정, 삭제할 수 있는 영역의 User Interface이다. 등록된 메뉴들도 활성화의 여부에 따라 메인화면에 Display 할 수 있다.



(그림.43) 관리자 페이지 - 메뉴 관리

- 메뉴 등록

메뉴 등록은 메인에 표시될 메뉴와 메인 메뉴 안에 표시될 사이드 메뉴로 구분되어 있으며, 원하는 항목을 선택하여 추가하면 Multi box list에 등록된다.

프로젝트명	자율주행 종합관제센터 표준 검증 플랫폼 개발_ UI 설계	화면명	관리자 메뉴 등록 페이지
화면경로	admin/menuadd.html	작성자	박시형
		작성일	2022-09-13

Description	
메인 페이지	
1	메뉴 등록 영역
1-1	메뉴 제목 기입 텍스트 창
1-2	메인 메뉴 Multi box
1-3	사이드 메뉴 Multi box
1-4	메뉴 등록 삭제 버튼

(그림.44) 관리자 페이지 - 메뉴 등록 화면

a. 배너 이미지 관리

배너 이미지를 첨부 파일을 통해 첨부 가능하며, 각 슬라이드별 등록, 삭제, 변경이 가능하다.

프로젝트명	자율주행 종합관제센터 표준 검증 플랫폼 개발_ UI 설계	화면명	관리자 배너 이미지 수정/등록 페이지
화면경로	admin/banneradd.html	작성자	박시형
		작성일	2022-09-13

Description	
메인 페이지	
1	배너 이미지 수정/등록 영역
1-1	각 배너 이미지에 대한 영역
1-2	파일업로드 및 저장 가능 버튼
1-3	전체 수정 후 저장 버튼

(그림.45) 관리자 페이지 - 배너 이미지 수정/등록

b. 공지사항 관리

메인화면에 표시될 대표 공지사항 글을 선택할 수 있으며, 대표 글로 쓰일 사진의 대표 이미지 또한 수정 가능하며, 공지사항의 대표 문구는 제목 혹은 설정에 따라 바꿀 수 있다.

프로젝트명	자율주행 통합관제센터 표준 검증 플랫폼 개발_ UI 설계	화면명	관리자 공지사항 수정/등록 페이지		
화면경로	admin/noticadd.html	작성자	박시형	작성일	2022-09-13

Description	
메인 페이지	
1	공지사항 수정/등록 영역
1-1	대표 공지사항 체크박스 리스트
1-2	대표 이미지 첨부 영역
1-3	메인페이지에 보여질 문구 작성 창구
1-4	전체 수정 후 저장 버튼

(그림.46) 관리자 페이지 - 공지사항 수정/등록

c. 표준 자료실 관리

표준 자료실도 공지사항과 같은 기능들이 있다.

프로젝트명	자율주행 통합관제센터 표준 검증 플랫폼 개발_ UI 설계	화면명	관리자 표준 자료실 수정/등록 페이지		
화면경로	admin/productadd.html	작성자	박시형	작성일	2022-09-13

Description	
메인 페이지	
1	표준 자료실 수정/등록 영역
1-1	대표 표준 자료 체크박스 리스트
1-2	대표 이미지 첨부 영역
1-3	메인페이지에 보여질 문구 작성 창구
1-4	전체 수정 후 저장 버튼

(그림.47) 관리자 페이지 - 표준 자료실 수정/등록

d. 관련 사이트 링크 관리

에디터를 통해 자유롭게 관련 사이트의 이미지를 첨부 및 삭제할 수 있다. 편집기에 나열한 사진 순서대로 링크의 이미지가 나열된다.



(그림.48) 관리자 페이지 - 관련 사이트 링크 수정/등록

ii. 유관기관 링크 관리

유관기관 링크도 관련 사이트와 같은 방법으로 수정/등록이 가능하다.



(그림.49) 관리자 페이지 - 유관기관 링크 수정/등록

4.1.9.6. 공지사항 관리

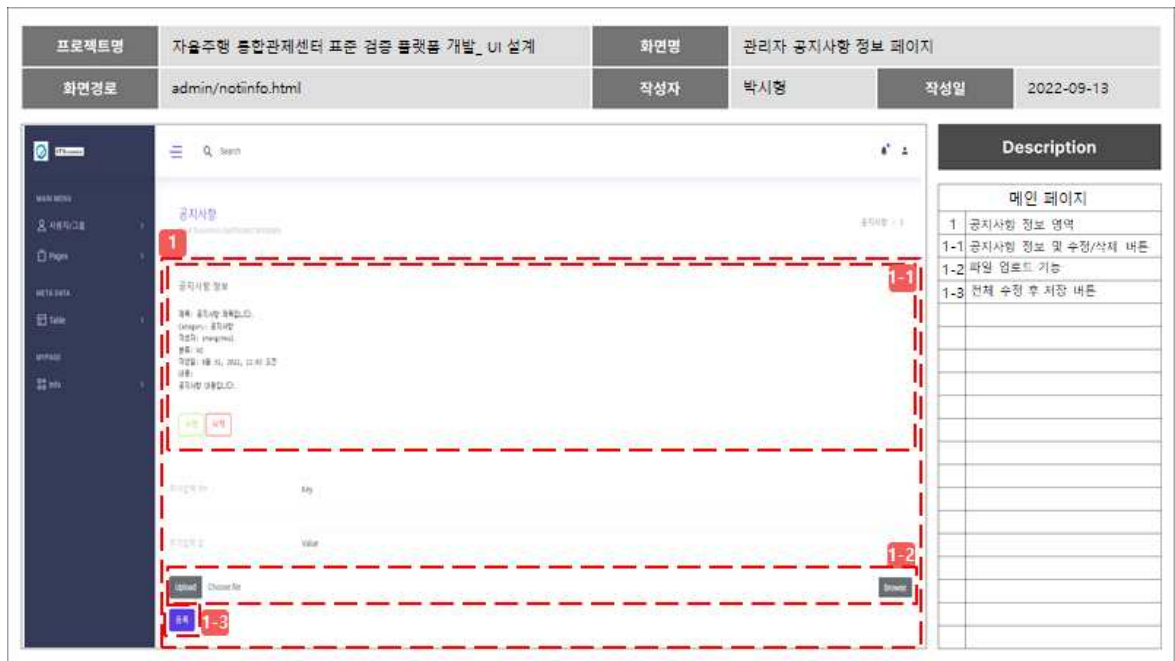
공지사항의 목록들을 확인할 수 있으며, 공지사항의 활성화 여부 또한 알 수 있다.



(그림 .50) 관리자 페이지 - 공지사항 관리

i. 공지사항 정보

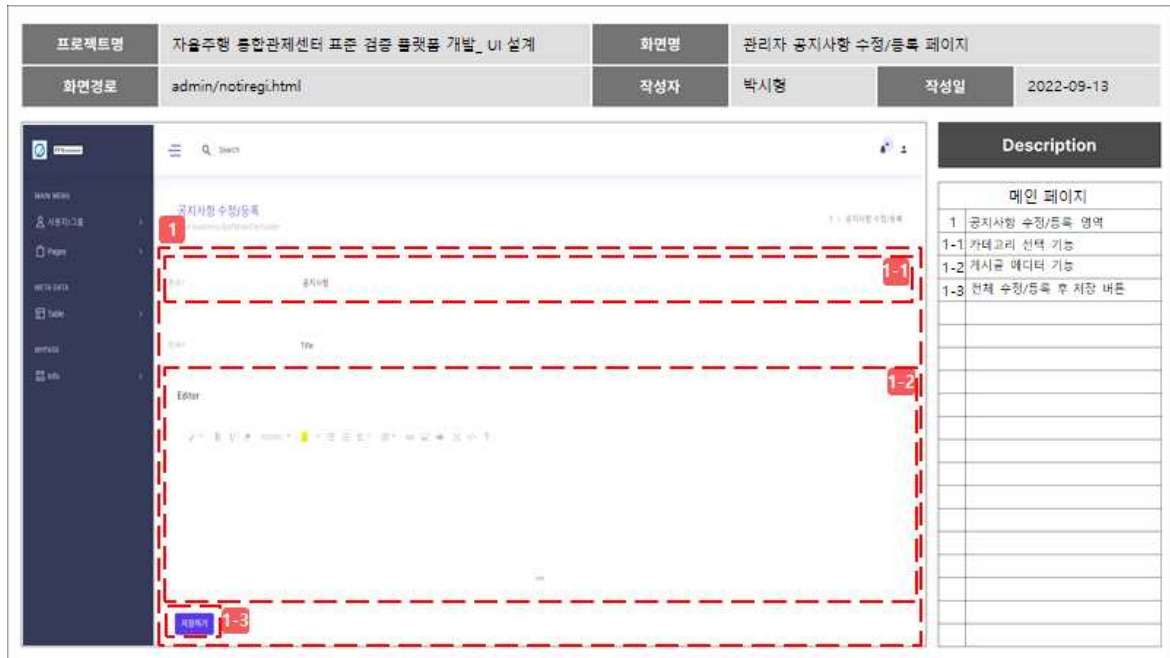
공지사항의 글을 수정 및 삭제할 수 있으며, 첨부 파일 업로드도 가능하다.



(그림 .51) 관리자 페이지 - 공지사항 정보

ii. 공지사항 수정/등록

공지사항을 에디터를 통해 수정/등록이 가능하며, 글 등록 시 사용자 페이지의 공지사항에 리스트업 할 수 있다.



(그림 .52) 관리자 페이지 - 공지사항 수정/등록

4.1.9.7. 표준 자료실 관리

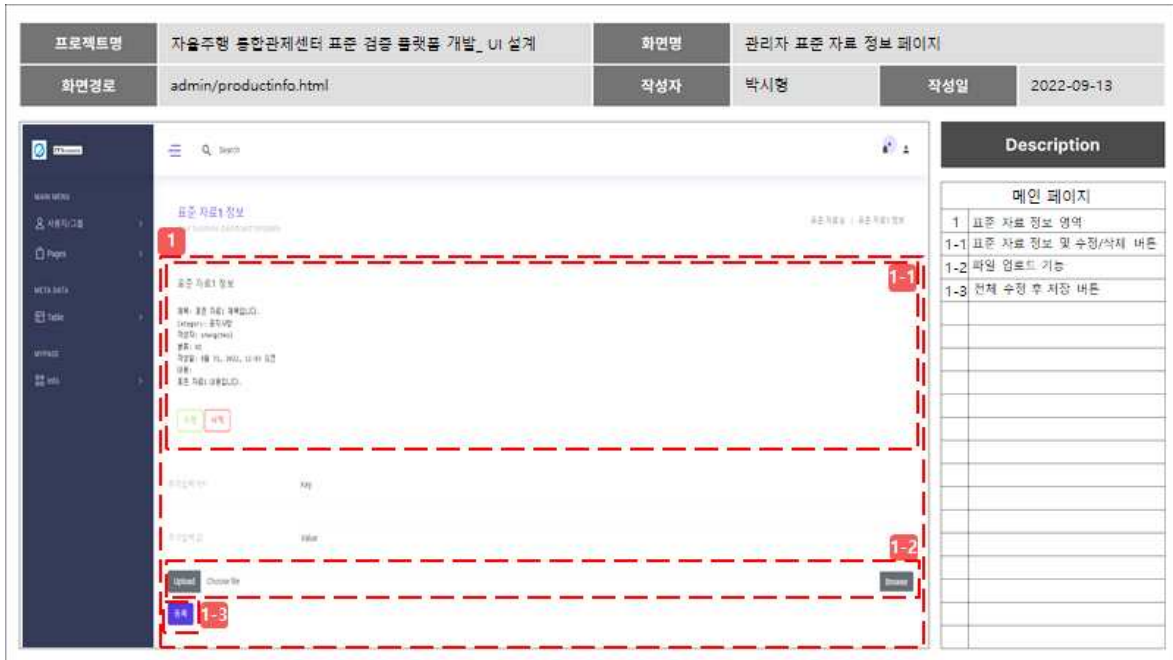
표준 자료들을 활성화할 수 있으며, 목록을 통해 표준 자료의 글 확인 및 수정, 등록이 가능하다.



(그림 .53) 관리자 페이지 - 표준 자료실 관리

a. 표준 자료 정보

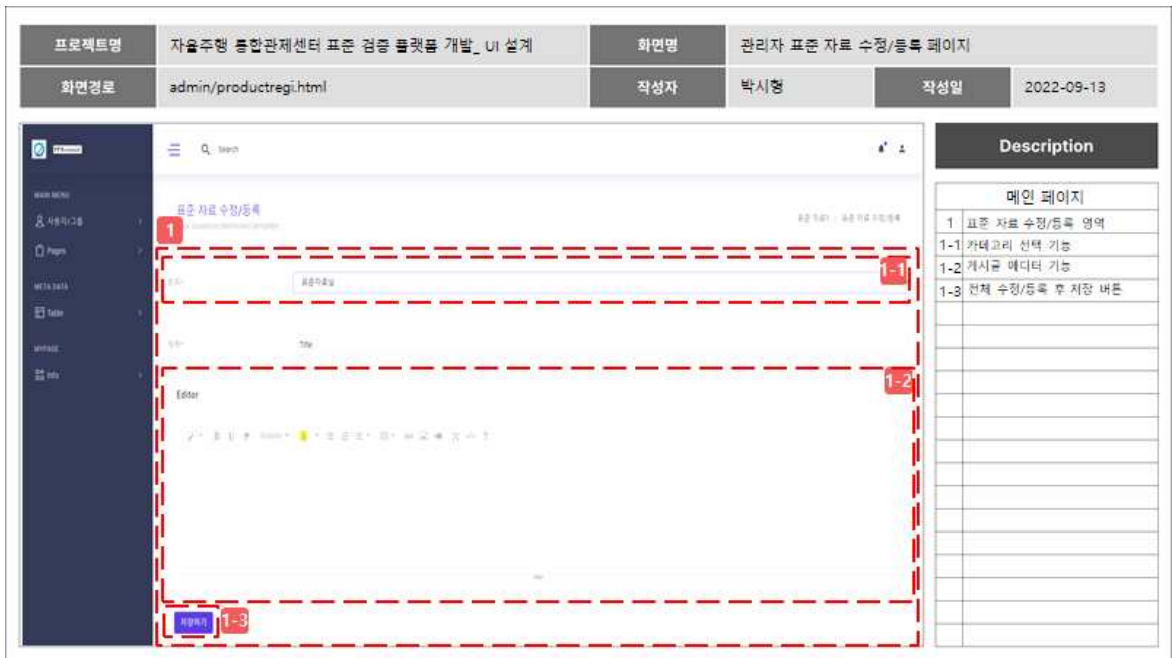
표준 자료의 정보를 열람할 수 있으며, 수정, 등록 및 첨부 파일 업로드가 가능하다.



(그림 .54) 관리자 페이지 - 표준 자료 정보

b. 표준 자료 수정/등록

표준 자료의 정보를 수정 및 등록할 수 있는 영역의 User Interface이다.



(그림 .55) 관리자 페이지 - 표준 자료 수정/등록

4.1.9.8. 카테고리 관리

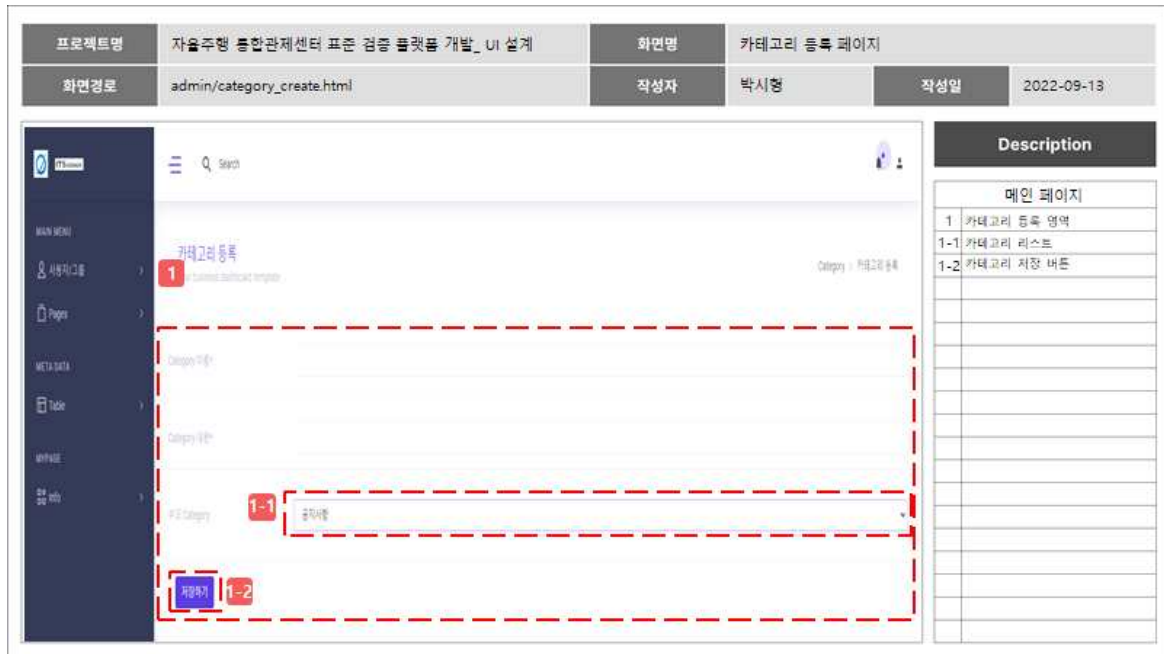
공지사항이나 표준 자료 혹은 게시글들을 등록할 때, 글의 종류를 고를 수 있는 카테고리를 관리하는 영역이다. 카테고리의 종류들을 확인할 수 있으며, 상단의 등록하기 버튼을 통해 카테고리 추가도 가능하다.



(그림 .56) 관리자 페이지 - 카테고리 관리

i. 카테고리 등록

카테고리를 등록할 수 있는 영역의 User Interface이다.



(그림 .57) 관리자 페이지 - 카테고리 등록

ii. 카테고리 정보

카테고리의 정보를 확인할 수 있으며, 해당 내용을 수정 혹은 삭제할 수 있다.



(그림 .58) 관리자 페이지 - 카테고리 정보

4.1.9.9. 메타데이터 관리

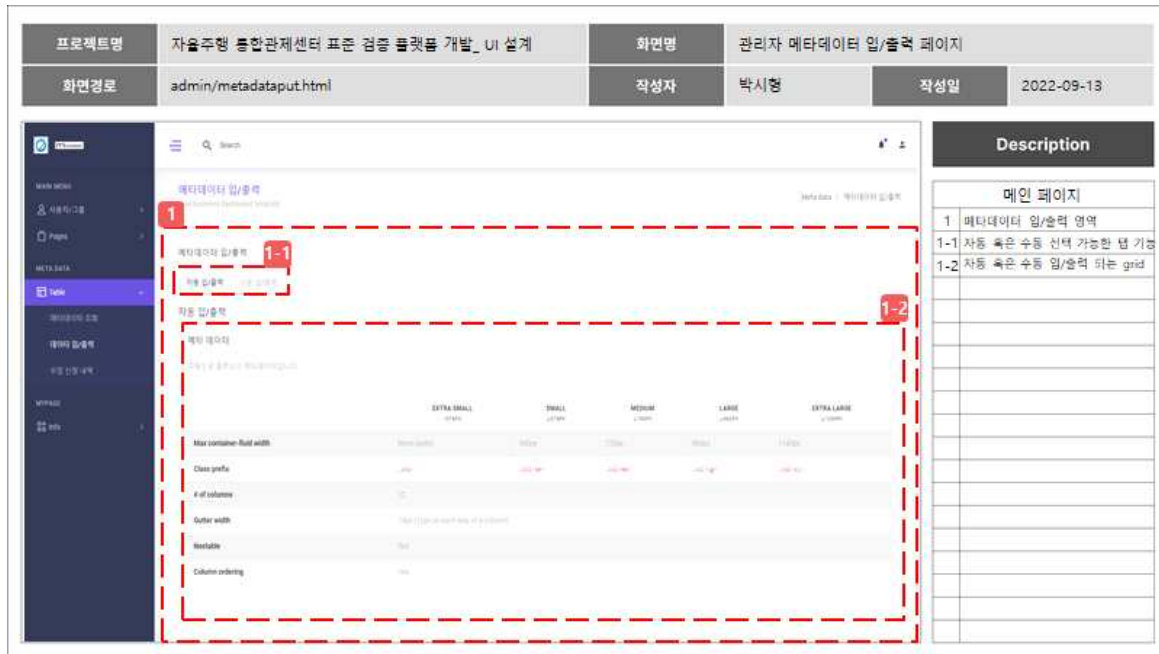
메타데이터를 조회할 수 있으며, 관리자는 자동 혹은 수동으로 표에 입/출력이 가능하다. 또한, 사용자들의 수정, 등록 신청 내용을 확인할 수 있고 승인 및 거절을 할 수 있다.



(그림 .59) 관리자 페이지 - 메타데이터 관리

i. 메타데이터 입/출력

관리자는 필요한 데이터를 수동으로 입/출력이 가능하며, 자동으로 데이터를 받아올 수 있다. 입/출력은 grid 형태로 받아온다.



(그림 .60) 관리자 페이지 - 메타데이터 입/출력

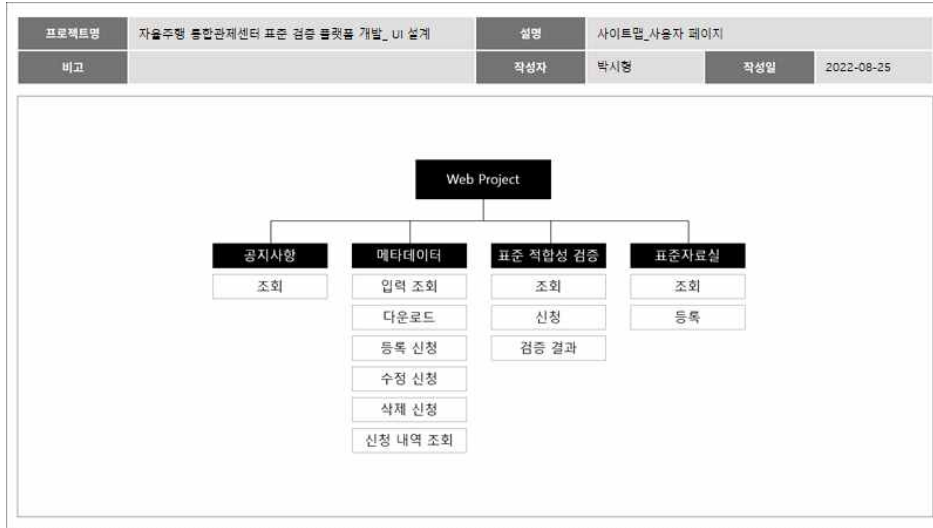
ii. 메타데이터 수정 신청 목록

관리자는 사용자가 신청한 목록들을 열람할 수 있으며, 승인, 거절 및 진행 중 세 가지로 구분 지어 표시할 수 있다.



(그림 .61) 관리자 페이지 - 메타데이터 수정 신청

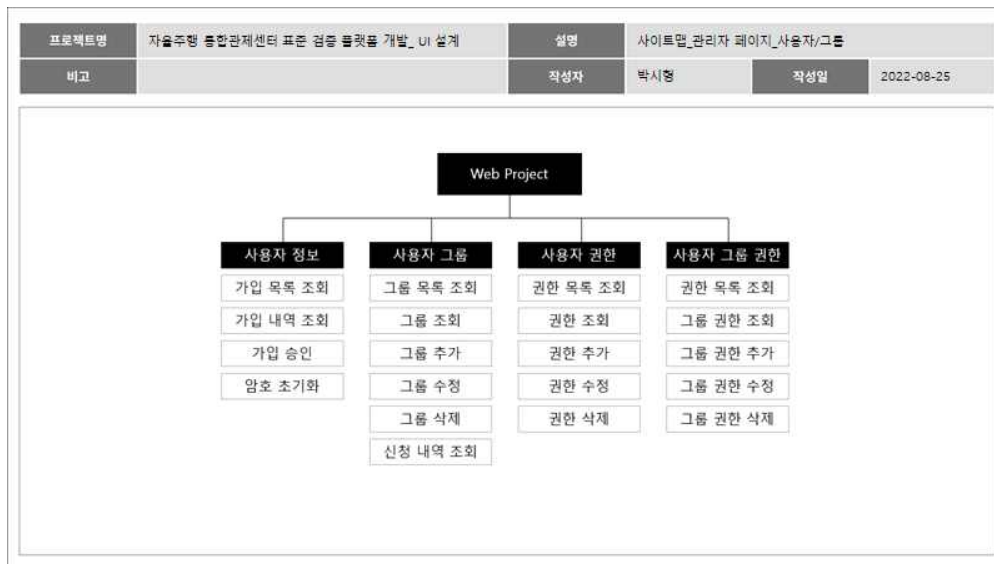
4.1.10. 사이트맵 설계



(그림 .62) 사이트맵 - 사용자 페이지

4.1.10.1. 사용자 페이지

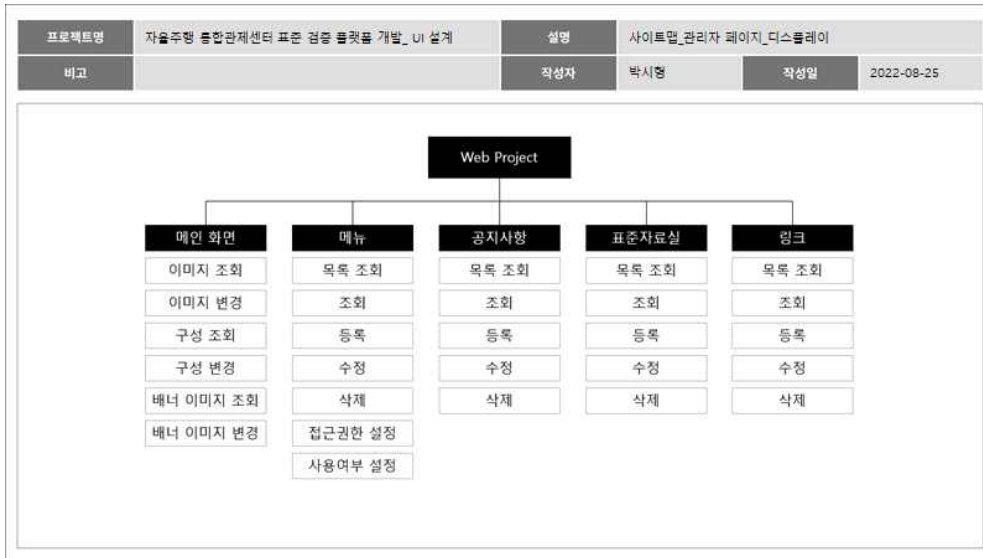
사용자 페이지의 중심이 되는 전체 메뉴들이며, 1차년도에 다루어질 메뉴는 공지사항, 메타데이터와 표준 자료실 기능이다.



(그림 .63) 사이트맵 - 관리자 페이지_사용자/그룹

4.1.10.2. 관리자 페이지의 사용자/그룹 관리

사용자의 정보를 관리하며 사용자가 가질 수 있는 그룹, 사용자의 권한 그리고 그룹의 권한을 Manage 할 수 있는 User Interface의 정리된 기능들이다.



(그림 .64) 사이트맵 - 관리자 페이지_디스플레이

4.1.10.3. 관리자 페이지의 디스플레이 관리

메인화면에 이미지들을 수정하고 변경할 수 있으며, 메뉴, 공지사항, 표준 자료들도 등록, 수정, 삭제 할 수 있다. WEB의 Update 할 수 있는 부분들을 관리자가 관리할 수 있는 User Interface이다.

5. SW 개발 환경

5.1. 메타데이터 관리시스템의 SW 개발 환경

○ Python

1991년에 발표된 인터프리터 방식의 프로그래밍 언어로 영어와 비슷해서 읽고 쓰기 쉬운 특유의 문법을 가지고 있고 그로 인해 전 세계의 개발자들로부터 많은 인기를 얻고 있다. 만들어진 수 많은 패키지들 덕분에 사용할 수 있는 용도가 무궁무진해서 2010년대 중반부터 주류 프로그래밍 언어로 널리 사용되고 있다.

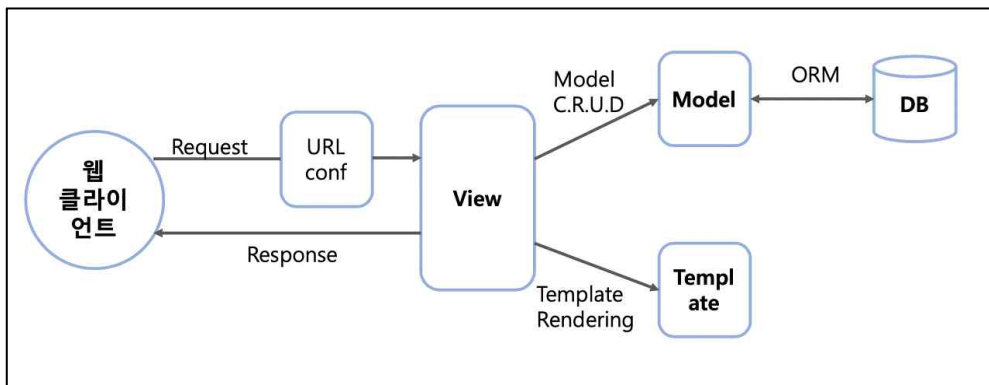
○ Conda

Python을 위한 가상환경으로 시스템에 설치되어 있는 python 버전과는 별도로 다수의 여러 버전 python을 설치 및 구동할 수 있다. 가상환경을 사용하는 주된 목적 중 하나는 개발 서버의 환경 구성과 상용 서버의 환경 구성을 동일하게 유지할 수 있으므로 개발과 디버깅이 용이하다는 점이다.

○ Python Django

Python Django는 python으로 만들어진 MVC(Model-View-Controller) 기반의 웹 프레임워크(Web framework)이며 django에서는 같은 개념을 MTV(Model - Template - View)라고 부른다.

MVC 패턴은 데이터(model), 사용자 인터페이스(view), 데이터 처리 로직(controller)을 구분해 하나의 요소가 다른 요소에게 영향을 주지 않도록 설계하는 방식이다. 아래 그림은 django의 MTV (Model - Template - View) 패턴에서 각 구성 요소의 상호 동작을 나타내는 그림이다.



(그림.65) python django 의 Model, View, Template 역할과 상호 연동 방법

Model은 MVC 패턴의 Model에 대응하는 개념이며 Template은 MVC 패턴의 View에 대응하는 개념으로 HTML 파일들을 통해 이용자가 시각적으로 확인할 수 있는 화면들을 보여주는 UI를 담당하는 부분이라고 볼 수 있다. 마지막으로 View는 MVC 패턴의 Controller에 대응하는 개념으로 웹 요청을 받고, 전달받은 데이터들을 해당 어플리케이션의 logic(로직)으로 가공한 후 그 결과를 Template에 전달하여 최종적으로 사용자에게 가공된 HTML 파일을 전달한다.

○ ORM

Object-relational mapping(객체 관계 매핑)의 약자로 데이터베이스의 테이블을 객체화하여, 각 DBMS에 대해서 CRUD(Create,Read,Update,Delete) 등을 공통된 접근기법으로 사용할 수 있다. 대표적인 Python ORM은 Django 자체적인 ORM 및 SQLAlchemy 등이 있다.

ORM의 장점으로 개발자는 DBMS에 대한 큰 고민 없이, ORM에 대한 이해만으로 쉽게 CRUD 연산을 다룰 수 있기 때문에, Business logic(업무 규칙)에 집중할 수 있으므로 개발 생산성을 증가시킬 수 있다. 또한 객체를 통하여 대부분의 데이터에 접근 및 수정을 하므로, 코드 가독성이 좋으며 데이터 구조 변경 시, 객체에 대한 변경만 이루어지면 되므로, 유지보수가 간편하다는 장점이 있다.

○ MySQL

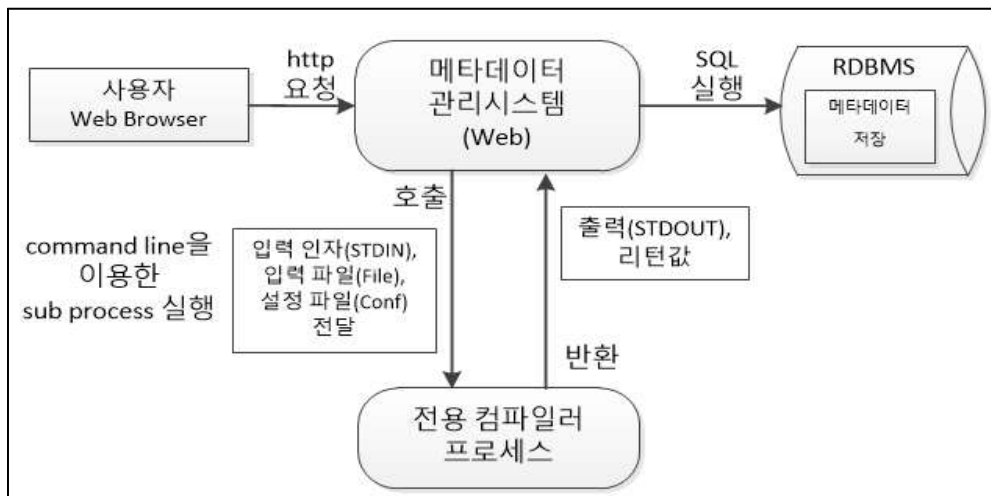
1995년에 발표된 오픈 소스 RDBMS로서 관계형 데이터베이스를 관리하는 시스템으로 테이블 행(컬럼) 단위로 검색, 삽입, 갱신, 삭제를 쉽게 해주는 관리 소프트웨어이다.

MySQL 5.7.8부터 DBMS 차원의 JSON 데이터를 지원하여 반정형(Semi-structured) 데이터를 테이블에 저장할 수 있게 되었다.

5.2. SW 연동 방안

5.2.1. 전용컴파일러 연동 방안

사용자가 전용컴파일러에 의한 메타데이터 추출을 하기 위해 메타데이터 관리시스템에 접속 및 추출 요청을 선택한 경우, 메타데이터 관리시스템은 해당 요청으로부터 사용자가 전달한 표준 문서(input)와 입력 인자(input parameter) 그리고 환경 설정 파일(configuration) 등을 command line 프로세스 호출을 위한 스트링으로 구성한 후 해당 스트링을 이용하여 sub process 방식으로 전용컴파일러를 호출한다.



(그림.66) 메타데이터 관리시스템과 전용컴파일러 간 호출 및 데이터 연동 방식

이때 메타데이터 관리시스템은 전용컴파일러에게 전달하는 입력 인자(입력 파일, 환경 설정 파일 포함)들을 command line parameter 전달 방식으로 해당 데이터들을 전달하고, 이후 전용컴파일러 프로세스는 메타데이터 추출을 완료한 경우 해당 메타데이터들을 STDOUT을 통해 출력 데이터를 메타데이터 관리시스템에게 전달한다. 이렇게 추출된 메타데이터들은 메타데이터 관리시스템에서 SQL 구문 실행을 통해서 RDBMS의 메타데이터 DB에 저장된다.

5.3. 데이터 운영 방안

5.3.1. 데이터의 분류

5.3.1.1. 정형 데이터(Structured Data)

정형 데이터는 구조화된 데이터, 즉 미리 정해진 구조에 따라 저장된 데이터다. 행과 열에 의해 지정된 칸에 데이터를 저장하는 엑셀의 스프레드 시트, 관계 데이터베이스의 테이블이 대표적인 예시이다.

5.3.1.2. 비정형 데이터(Unstructured Data)

비정형 데이터는 정해진 구조가 없이 저장된 데이터다. 일반적인 텍스트, 이미지, 영상, 워드나 PDF 문서와 같은 멀티미디어 데이터가 대표적인 예다.

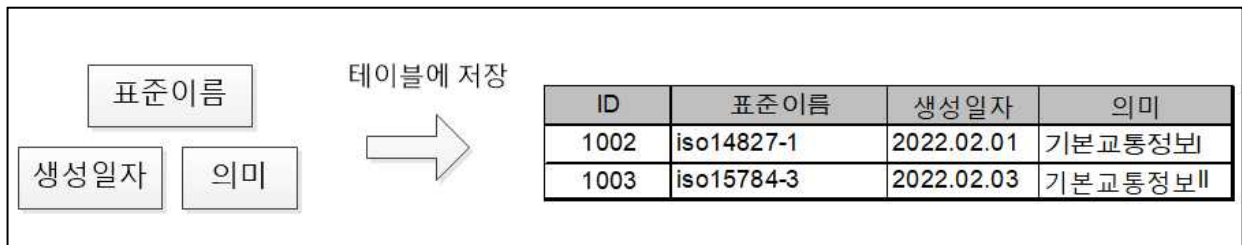
5.3.1.3. 반정형 데이터(Semi-Structured Data)

반정형 데이터는 특정 구조에 따라 저장된 데이터지만 정형 데이터와 달리 보통 데이터 내용 안에 구조에 대한 설명이 함께 존재한다. 따라서 데이터 내용에 대한 설명인 구조를 파악하는 분석 과정이 필요하고, 보통 파일 형태로 저장된다. 반정형 데이터의 예시로 웹에서 데이터를 교환하기 위해 작성하는 HTML, XML, JSON 문서나 웹 로그 등이 있다.

5.3.2. 메타데이터 관리시스템의 데이터 운영 방안

5.3.2.1. 정형 데이터(Structured Data) 운영 방안

표준번호, 표준이름, 데이터 생성 일자, 데이터 의미 등의 메타데이터 관리 정보에 해당하는 데이터는 메타데이터 추출(혹은 사용자의 입력) 및 저장 시 생성되는 공통적인 정보이다. 따라서 이러한 데이터의 저장은 일반적인 테이블의 고정 컬럼(Column)을 이용하여 저장하고 이후 수정 및 검색 시 해당 컬럼의 값을 이용하여 연결된 데이터를 추출 및 수정할 수 있도록 한다.

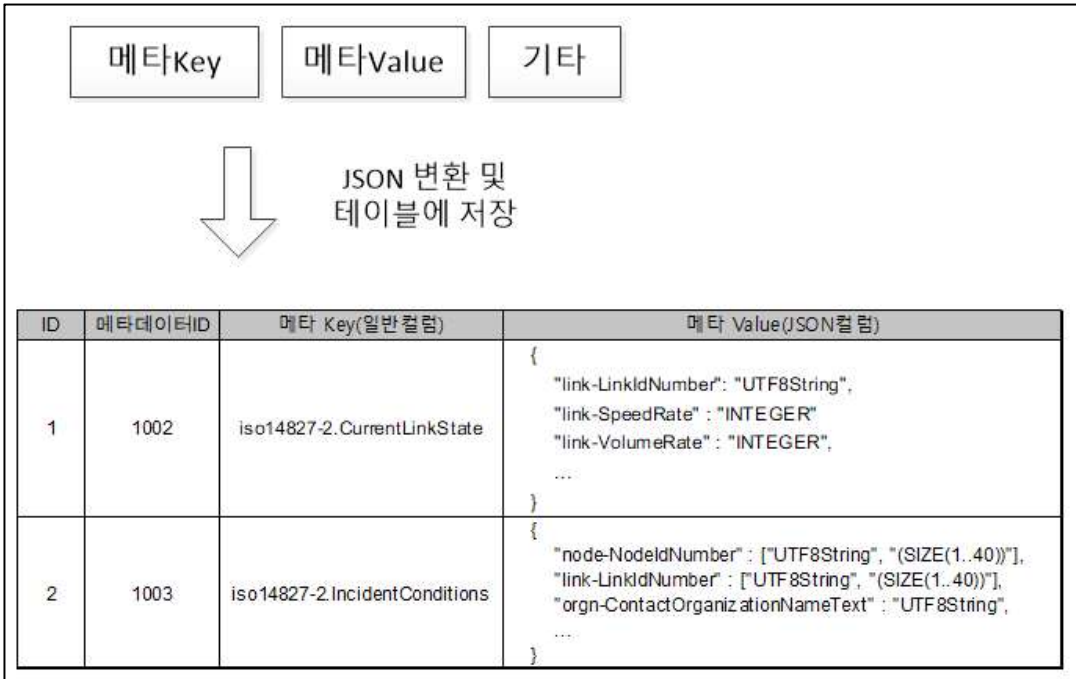


(그림.67) 정형 데이터의 테이블 저장 방식

5.3.2.2. 반정형 데이터(Semi-Structured Data) 운영 방안

전용컴파일러에 의해 추출된 메타데이터(혹은 사용자가 수동으로 입력한 메타데이터)는 크기와 데이터 구조를 미리 알 수 없는 반정형 데이터로 볼 수 있으므로 테이블의 고정 크기 컬럼에 저장될 수 없다.

DBMS 중 하나인 MySQL 5.7.8부터 DBMS 차원에서 JSON 데이터를 지원하여 반정형(Semi-structured) 데이터를 테이블에 저장할 수 있게 됨에 따라, 전용컴파일러에 의해 추출된 메타데이터의 경우 테이블의 JSON 컬럼(Column)을 생성 및 이용하여 해당 데이터를 저장하고 이후 검색 및 수정 시 MySQL의 JSON 함수를 이용하여 해당 데이터를 검색 및 수정할 수 있도록 한다



(그림 .68) 반정형 데이터의 테이블 저장 방식

6. 인터페이스

인터페이스 모듈은 Protocol Engine이 접근할 수 없는 System Resource나 Network Device, 또는 특정 알고리즘을 사용하기 위해 필요한 보조 모듈이다.

6.1. 기능 설계

6.1.1. Profiling 기능

인터페이스 모듈은 미지의 기능이나 장치를 사용할 수 있게 해주는 모듈로서, 설계 시점에서 명확한 기능을 특정할 수 없다. 따라서 Runtime에 확정된 기능이 어떤 것이 있는지 Governor에게 알려줄 방법이 필요하며, 이를 Profiling 기능을 통해 구현한다.

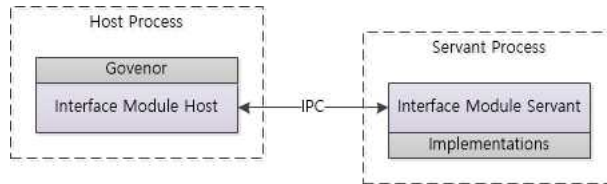
Profiling 정보에는 다음과 같은 내용을 포함한다.

- Registry Profiling
모듈이 사용하는 Data 집합을 Profiling 한다. 단일 변수, 자료 구조체 및 List에 관한 정보를 포함하며, Meta Data 표기법을 사용하여 표현한다.
- Function Profiling
모듈이 제공하는 기능 집합을 Profiling 한다. 기능의 명칭과 인수로 사용할 Data 구조 및 결과 Data의 구조를 포함하며 Meta Data 표기법을 사용하여 표현한다.
- Control Profiling
모듈을 제어할 수 있는 항목 집합을 Profiling 한다. 기본적으로 Function Profiling과 동일하지만, 동작 특성이 다를 수 있는 이유로 별도 정의한다.

6.1.2. Functionality Invocation 기능

Profiling에 의해 정의된 Registry, Function 및 Control을 동작시키는 기능이다. Registry의 경우 Read, Write 기능을 수행하고, Function 및 Control의 경우 Execute 기능을 수행한다.

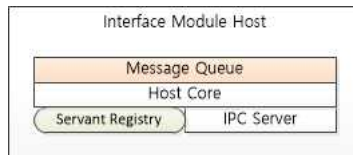
6.2. 블록 설계



(그림.69) Interface Module

Interface Module은 Host와 Servant로 구분된다. Host는 Interface Module을 사용하고자 하는 Governor가 존재하는 Process에 위치한다. Servant는 Interface 기능을 구현한 Process에 존재한다.

6.2.1. Interface Module Host

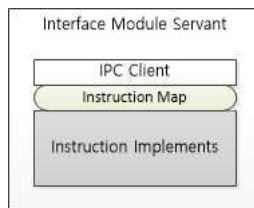


(그림.70) Interface Module

Host

Host Module은 Governor의 제어를 수신하기 위한 Message Queue, 수신한 Message에 따라 동작을 수행하는 Code, Servant와 연동하기 위한 IPC(Inter-Process Communication) Server, 그리고 Servant를 관리하기 위한 Servant Registry로 구성된다.

6.2.2. Interface Module Servant



(그림.71) Interface

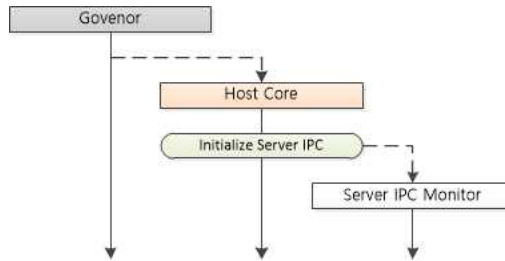
Module

Servant

Servant Module은 Host와 통신하기 위한 IPC Client, Servant Module이 제공하는 기능을 수행하기 위한 Instruction Map, 그리고 각 Instruction에 대한 구현내용으로 구성된다. Instruction에 대한 구현 내용은 실제 개발 시 Instruction Map의 내용에 따라 결정된다.

6.3. 구현 설계

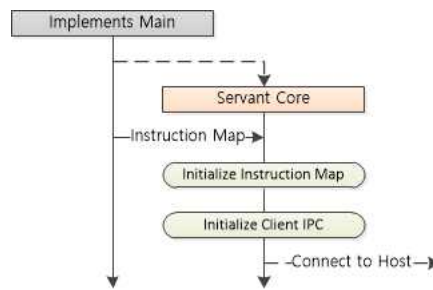
6.3.1. Host 초기화 기능



(그림.72) Host Module Initialize

Host Module을 사용하는 Govenor가 Host Module을 생성한다. 생성된 Host Module의 Core Thread는 Server IPC를 생성하고 Monitoring Thread를 생성하여 Servant Proccoss의 접속을 기다린다.

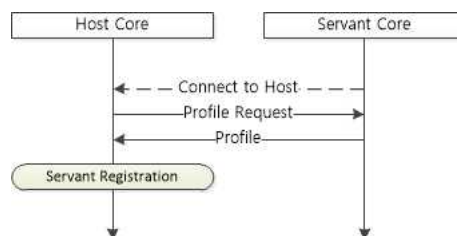
6.3.2. Servant 초기화 기능



(그림.73) Servant Module Initialize

Servant Interface Module을 사용하는 Implements Main이 Servant Module을 생성한다. 생성된 Servant Module의 Core Thread는 Implements Main이 전달하는 Implement Map을 기다린다. Servant Core Thread가 Implement Map을 수신하면 Client IPC를 생성하여 Host Module로 접속을 시도한다.

6.3.3. Establishment & Negotiation

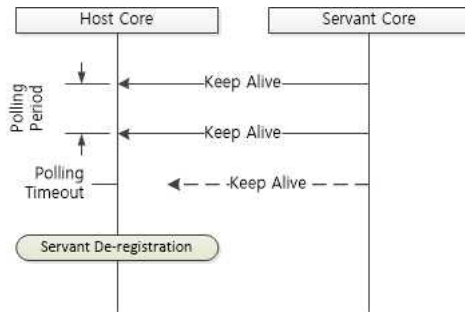


(그림.74) Establishment & Negotiation

Host Module이 Servant Module의 접속을 감지하면 Profile Request Message를 Servant Module로 전송한다. Servant Module이 Profile Request Message를 수신하면 Instruction Map을 기반으로

Profile을 생성하여 Host Module로 전송한다. Host Module이 Profile을 수신하면 수신한 Profile을 이용하여 Servant Entry를 생성하고, Servant Registry에 등록한다.

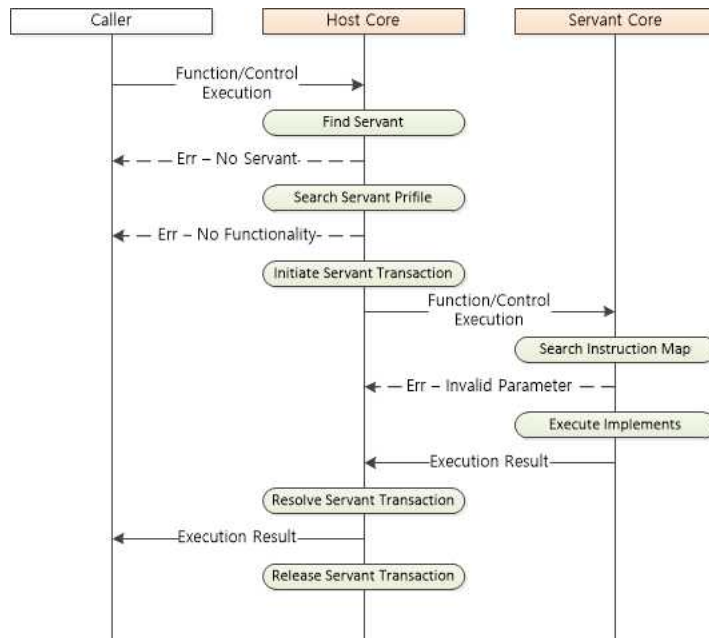
6.3.4. Polling



(그림.75) Polling

Servant Module이 Profile을 전송한 이후 일정 시간 동안 Host와 Message 교환이 없을 경우 Keep Alive를 Message를 Host Module로 전송한다. Host Module은 Keep Alive Message를 포함한 어떠한 Message라도 수신한 경우 Keep Alive Timer를 갱신한다. Host Module이 Servant Module로부터 일정시간 이상 아무런 Message도 수신하지 못하면 해당 Servant의 정보를 Registry에서 제거하고 연결된 IPC Session을 제거한다.

6.3.5. Functionality Invocation



(그림.76) Functionality Invocation

Caller는 필요한 기능 또는 제어를 하기 위한 Function/Control Execution Message를 Host Module로 전달한다. Host Core는 해당 Message를 처리할 Servant를 Registry에서 검색한다. (찾지 못한 경우 No Servant Error를 돌려준다.) Servant Module이 검색되면 해당 Servant에 해당 Message에서 요구한 기능 또는 제어가 있는지 검색한다. (없는 경우 No Functionality를 돌려준다.) Host Module은 Caller 정보와 검색된 Servant Module을 저장한 Transaction을 생성하고, Servant Module에게 Function/Control Execution Message를 IPC를 통해 전달한다. Servant Module은 요구한 Function/Control을 Instruction Map에서 검색한다. (검색된 Instruction에 필요한 Parameter가 적절치 않은 경우 Invalid Parameter를 돌려준다.) Servant Module은 검색된 Implementation을 실행하고 결과를 Host Module에게 IPC를 통해 전달한다. Host Module이 결과를 수신하면 Transaction Registry에서 해당 Transaction을 검색하여 Caller를 찾는다. Host Module이 Caller를 찾으면, 결과를 Caller에게 전달하고 Transaction은 삭제한다.

6.4. 상세 설계

6.4.1. IPC Frame

GUID dst_module_id	GUID src_module_id	uint16 pkt_seq	uint16 contents_type	uint32 contents_size	ByteArray contents_data
-----------------------	-----------------------	-------------------	-------------------------	-------------------------	----------------------------

(그림 .77) IPC Frame

- dst_module_id
Data를 수신할 Module의 ID. GUID 타입
- src_module_id
Data를 송신하는 Module의 ID. GUID 타입
- pkt_seq
송신 프레임 번호. 0부터 시작하며 65535까지 증가하고, 다시 0으로 리셋된다.
- contents_type
contents_data에 저장된 정보의 종류. 구체적인 종류는 개발을 진행하면서 확정한다.
- contents_size
contents_data의 길이. (단위 Octet)
- contents_data
contents_type에 따른 Data. 구체적인 내용은 개발을 진행하면서 확정한다.

6.4.2. Profile Syntax

Profile Syntax를 Interface Module 설계 단계에서 완전히 확정 지을 수는 없지만 다음과 같은 사항을 기본적으로 따른다.

- 모든 Data Type의 정의는 Protocol Engine 설계의 Message Spec을 따른다.
- 모든 Data Instance의 정의는 Protocol Engine 설계의 Protocol Data Description을 따른다.
- 모든 Function 및 Control의 정의는 이름과 Parameter Data Type, Result Data Type을 포함하여야 하며, 각각의 Data Type 정의는 Protocol Engine 설계의 Message Spec을 따른다.

7. 시나리오 엔진

시나리오 엔진은 시험 대상 시스템과 통신하여 시험 대상이 메시지를 적절하게 생성하여 전송하고, 시험 대상으로 전송한 메시지를 적절하게 해석하는지 확인하는 모듈이다. 하지만 프로토콜은 다양한 형태로 존재하고, 송수신 정보는 프로토콜에 따라 다양하다. 시나리오 엔진은 특정 프로토콜에 종속되지 않도록 구현해야 함으로, 프로토콜 동작은 실행 시에 결정될 수 있도록 설계한다.

시나리오 엔진은 하드웨어 및 OS와 직접적인 연관 관계가 없는 Application Layer의 Protocol을 다룬다. Application Layer는 1개 이상의 계층구조로 구성될 수 있고, 시나리오 엔진은 한번에 하나의 Layer만 다룰 수 있기 때문에, 완전하게 구현한 하위 Layer를 실행시에 Binding 할 수 있어야 한다. 또한 시나리오 엔진이 Peer의 동작을 확인하려면 시나리오 엔진이 시험하고자 하는 Protocol이 Peer 시스템의 최상위 Application Protocol이어야 한다.

7.1. 기능 설계

7.1.1. Protocol Spec Loading 기능

Protocol Spec을 Governor로부터 제공받아 Loading하는 기능이다. 본 기능은 다음과 같은 Spec을 Loading한다.

- Protocol Message Spec.
Protocol이 사용하는 Message에 대한 Meta 정보
- Message Capsulation Rule.
다층 구조의 Message 생성 시 상호 포함 관계 및 Embedding 방법
- Specific Message Response Rule
Polling형 Message에 대한 응답 방법 및 타이머에 의한 Message 생성 방법

7.1.2. Upper Layer Interface 기능

시나리오 엔진은 상위 레이어와 연동하기 위해 다음과 같은 기능을 제공해야 한다.

- Drop Down Message
상위 레이어로부터 Message를 받아 주어진 Protocol Spec에 따라 처리하는 기능이다.
- Push Up Message
상위 레이어로 Message를 전송하는 기능이다. 하위 레이어에서 수신한 Message를 Protocol Spec에 따라 처리하는 과정에서 사용하는 기능이다.

7.1.3. Lower Layer Interface 기능

시나리오 엔진은 하위 레이어와 연동하기 위해 다음과 같은 기능을 제공해야 한다.

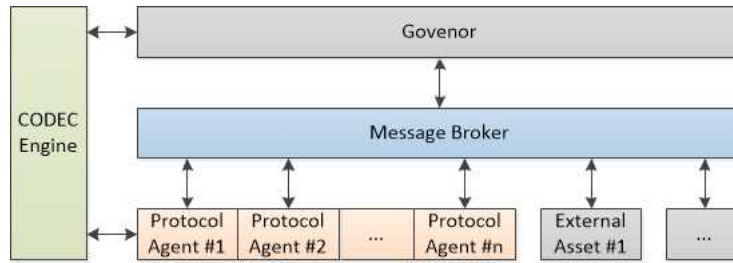
- Sink Down Message
하위 레이어로 Message를 전송하는 기능이다. 상위 레이어에서 수신한 Message를 Protocol Spec에 따라 처리하는 과정에서 사용하는 기능이다.
- Pop Up Message
하위 레이어로부터 Message를 받아 주어진 Protocol Spec에 따라 처리하는 기능이다.

7.1.4. Control 기능

시나리오 엔진의 동작에 영향을 미치는 여러 가지 제어 명령에 대한 동작을 수행하는 기능이다. 상위 레이어로부터 수신한 Command Message에 대한 처리와 하위 레이어로부터 수신한 Notify Message에 대한 처리를 담당한다.

- Registry Access
Registry는 시나리오 엔진의 동작에 사용하는 각종 Parameter를 저장하는 공간이며, 상위 레이어 및 하위 레이어는 Registry를 Access하여 Negotiation 및 Processing 방식을 변경한다.
- Command 실행 기능
Protocol Scenario상 나타날 수 없는 동작을 수행해야 할 필요가 있는 경우(접속 강제 종료 등) Command Message를 정의하고 이를 시나리오 엔진에 전달하여 구현하여야 한다. 이때 사용되는 Command Message에 대한 동작을 수행하는 기능이다.

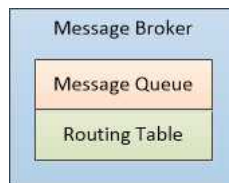
7.2. 블록 설계



(그림.78) Protocol Engine

Protocol Engine은 크게 세 개의 Block으로 구성된다. Message Broker는 Governor와 Protocol Agent 사이에서 Message 전달을 담당한다. Protocol Agent는 Application Protocol Simulation을 담당한다. 하나의 Protocol Agent는 하나의 Application Protocol Layer를 담당한다. CODEC Engine은 Governor 및 Protocol Agent가 사용하는 Message를 Stream으로 ENCODING 하거나 Stream을 Message로 DECODING 하는 역할을 담당한다.

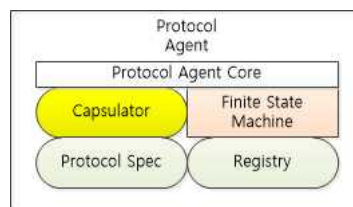
7.2.1. Message Broker



(그림.79) Message Broker

Message Broker는 메시지 전달을 위한 Resource로 Message Queue와 Routing Table을 가지고 있다. Message Queue는 Governor 및 Protocol Agent, External Asset으로부터 수신한 Message를 차례대로 저장하는 저장 공간이다. Routing Table은 Message에 지정된 Source와 Destination을 참고하여 목적지(Agent나 Asset, 또는 Governor)를 결정한다.

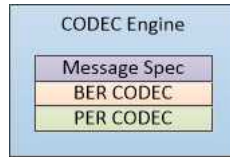
7.2.2. Protocol Agent



(그림.80) Protocol Agent

Protocol Agent는 Message Broker로부터 수신한 Message를 수신하거나 처리하기 위해 Agent Core, Protocol Spec, Finite State Machine 및 Registry를 가지고 있다. Agent Core는 Message Broker로부터 수신한 Message를 차례대로 저장하고 Protocol Spec에 지정한 대로 처리한다. Protocol Spec은 수신한 Message의 내용에 따라 수행할 동작을 기술한 Meta Data의 집합이며, Protocol Agent가 생성될 때 주어진 내용을 저장하는 저장 공간이다. Registry는 Protocol Spec을 분석하여, Protocol Spec에 주어진 동작에 필요한 내부 변수와 데이터를 저장하는 저장 공간이다.

7.2.3. CODEC Engine

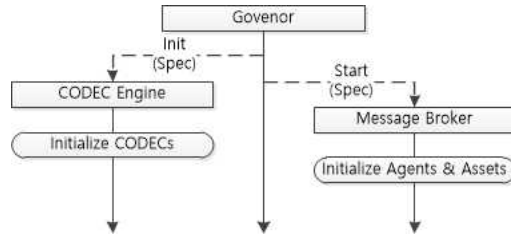


(그림 .81) CODEC Engine

CODEC Engine은 Govenor 및 Protocol Agent로부터 수신한 Message나 Stream을 특정 CODEC을 이용하여 Encode 하거나 Decode한다. 이를 위해 Meta Data로 구성된 Message Spec을 저장하는 저장 공간과 Meta Data를 이용하여 Encode/Decode를 할 수 있는 CODEC(BER, PER)을 가지고 있다.

7.3. 구현 설계

7.3.1. 초기화 기능

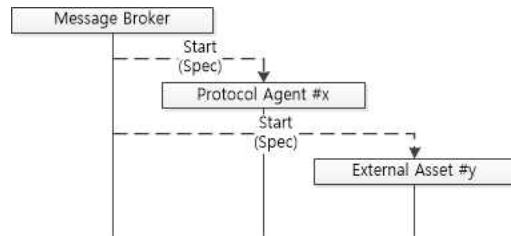


(그림 .82) 초기화 기능

Governor는 Protocol Spec 및 Message Spec을 수급받아 CODEC Engine과 Message Broker를 생성한다. 생성된 CODEC Engine은 Message Spec을 해석하여 Encoder 및 Decoder를 초기화 한다. 생성된 Message Broker는 Protocol Spec을 해석하여 Protocol Agent 및 External Asset을 생성한다.

7.3.2. Message Broker

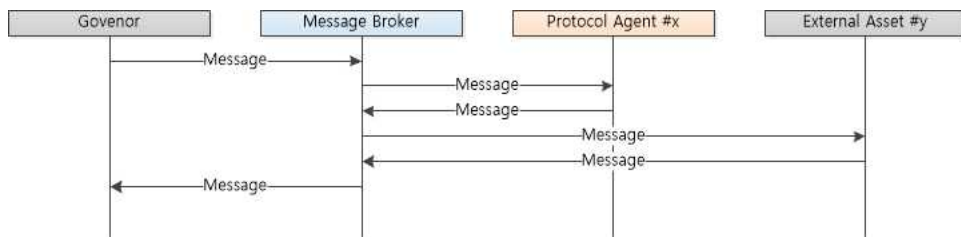
7.3.2.1. 초기화 기능



(그림 .83) Message Broker 초기화 기능

Message Broker는 주어진 Protocol Spec을 해석하여 Spec이 기술된 Protocol Stack에 따라 각 Layer별 전담 Agent를 생성하고, 필요시 External Asset도 실행 시킨다.

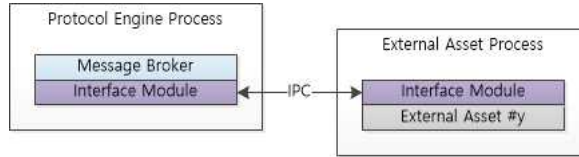
7.3.2.2. Message Routing 기능



(그림 .84) Message Routing 기능

Message Broker는 주어진 Protocol Spec을 해석하여, Layer와 Protocol Agent의 Mapping 정보를 생성하고, 각 Layer 간의 상하 관계를 파악하여 Protocol Message Flow에 따른 Message 전달 경로를 생성한다. 이후 Governor 및 External Asset으로부터 수신된 Message는 이 전달 경로에 따라 Protocol Agent 및 Governor, External Asset 등으로 전달한다.

7.3.2.3. External Asset 연동 기능

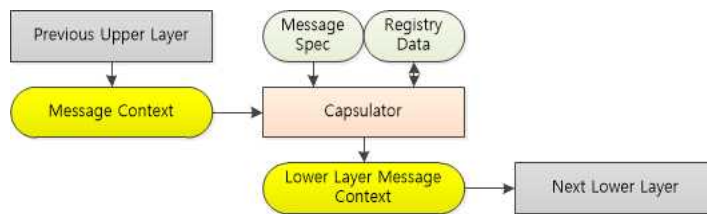


(그림 .85) External Asset 연동 기능

External Asset은 Protocol Engine이 수용할 수 없는 독특한 동작의 Protocol이나, 전형적인 Socket Interface를 따르지 않는 전용의 통신 수단을 이용해야 할 때, 전용의 구현물(Process 또는 SDK, Library 등)을 이용하여 Protocol Engine과 연동 할 수 있게 만든 Process이다. Message Broker는 Interface Module 설계에 정의한 방법론을 따라 연동하게 구현하고, 동일한 방법론을 따라 구현한 외부 모듈을 External Asset으로 인식하여 연동한다.

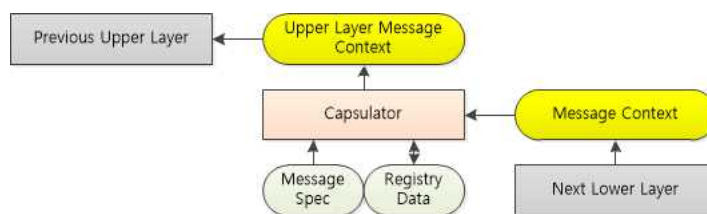
7.3.3. Protocol Agent

7.3.3.1. Capsulation 기능



(그림 .86) Encapsulation

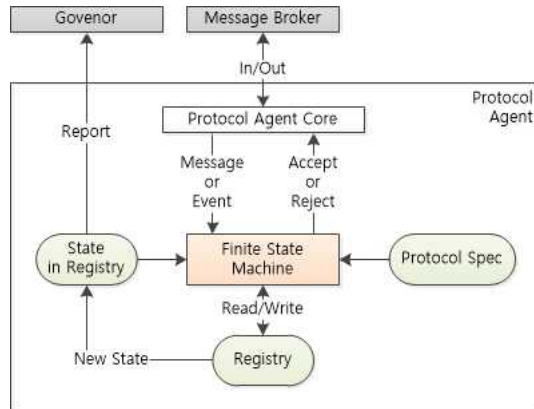
Protocol Spec에 의해 Layer의 Message와 Lower Layer의 Message의 구조가 결정된다. 이때 Layer Message 및 Registry에 저장된 Data가 Lower Layer Message에 어떻게 연관되는지 인식 할 수 있으며, Capsulator는 이 정보에 따라 Lower Layer Message를 정상적으로 생성하는 역할을 담당한다.



(그림 .87) Decapsulation

Protocol Spec에 의해 Layer의 Message와 Upper Layer의 Message의 구조가 결정된다. 이때 Lower Layer에서 공급된 Message 중 어떤 것이 Registry에 반영될지, Upper Layer로 전달될지 인식 할 수 있으며, Capsulator는 이 정보에 따라 Layer Message를 분리하여 Upper Layer Message를 생성하는 역할을 담당한다.

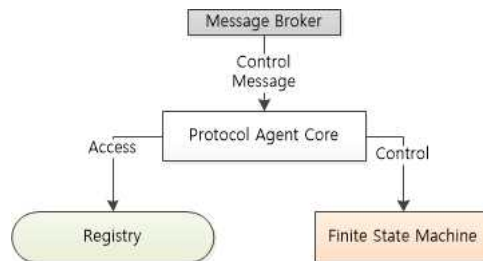
7.3.3.2. Finite State Machine 기능



(그림 .88) Finite State Machine

대부분의 Protocol은 상태에 따라 송신 및 수신 가능한 Message가 제한될 수 있는 형태를 가지고 있다. 이를 위해 Layer의 상태를 추적하는 기능 모듈이 필요하며, Finite State Machine이 해당 기능을 담당한다. 상태의 정의와 변경규칙은 Protocol Spec에서 추출한다.

7.3.3.3. Control 기능

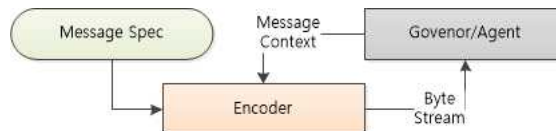


(그림 .89) Protocol Agent Control

Protocol은 Activation, De-activation 및 Termination 등의 Exception Case를 가지고 있다. 이를 Protocol Engine이 수용하기 위해 제어 명령을 전달할 수 있는 방법을 제공하고, Agent Core는 수신한 제어 명령에 따라 Registry의 값을 변경하거나 Finite State Machine의 상태를 강제로 제어하는 등의 기능을 담당한다.

7.3.4. CODEC Engine

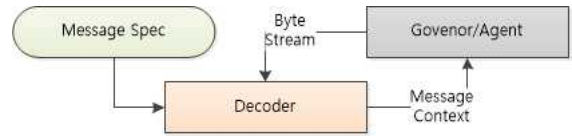
7.3.4.1. Encode 기능



(그림 .90) Encoder

모든 Protocol은 Peer Layer로 Message를 전송하기 위해 Layer Message를 직렬화 된 Byte Stream으로 변환해야 한다. 이를 위해 CODEC Engine의 Encoder는 Message Spec을 기반으로 Governor 및 Protocol Agent가 전달한 Message Context를 Byte Stream으로 변환하는 기능을 담당한다. 이때 Context와 Stream의 상호 변환 관계는 Encoding Rule이라는 정의가 필요하다. CODEC Engine은 X.690 Basic Encoding Rule과 X.691 Packed Encoding Rule을 지원한다.

7.3.4.2. Decode 기능



(그림 .91) Decoder

모든 Protocol은 Peer Layer로 Message를 수신하기 위해 직렬화 된 Byte Stream을 Layer Message로 변환해야 한다. 이를 위해 CODEC Engine의 Decoder는 Message Spec을 기반으로 Governor 및 Protocol Agent가 전달한 Byte Stream을 Message Context로 변환하는 기능을 담당한다. 이때 Context와 Stream의 상호 변환 관계는 Encoding Rule이라는 정의가 필요하다. CODEC Engine은 X.690 Basic Encoding Rule과 X.691 Packed Encoding Rule을 지원한다.

7.4. 상세 설계

7.4.1. Protocol Spec

Protocol Spec을 Protocol Engine 설계 단계에서 완전히 확정 지을수는 없지만, 다음과 같은 범주의 항목이 포함되어야 한다.

7.4.1.1. Message Spec

Message Spec은 Message Context에 대한 Meta-Data이며, Message Context의 구조와 각 Member의 유형 및 크기를 기술하고 있어야 한다. Message Spec은 ASN.1 Compiler와 같은 Tool을 통해 Abstract Syntax Notation 1 규격으로 기술한 Message Context를 생성 할 수 있으며, Protocol Spec에 사용되는 규격에 맞춰 수동으로 작성 할 수도 있다.

Message Spec에 사용되는 Primitive Type은 다음과 같다.

- Boolean
참과 거짓을 표현하는 유형
- Integer
정수(음수값 포함)값을 표현하는 유형
- Bit String
Bit Flag 배열을 표현하는 유형
- Octet String
Binary Data를 표현하는 유형
- NULL
값이 지정되지 않는 유형
- Object Identifier
OID를 표현하는 유형
- Real
실수값을 표현하는 유형
- Enumerated
열거형 값을 표현하는 유형
- UTF-8 String
UTF-8 형식으로 문자열을 표현하는 유형
- Relative Object Identifier
OID를 표현하는 유형, 단 특정 기준 OID로부터 다른 부분만 표현하는 유형
- Generalized Time
연월일시분초를 표현하는 유형
- Universal String
UCS-4BE 형식으로 문자열을 표현하는 유형
- BMP String
UBS-2BE 형식으로 문자열을 표현하는 유형

또한, 위의 Primitive Type은 다른 이름으로 재 정의하여 고유한 의미를 가지는 유형을 정의할 수 있다.

- SEQUENCE / SET
다수의 Primitive Type 또는 중첩된 SEQUENCE / SET 유형을 사용하여 구조화된 계층형 정보를 표현하는 유형

- CHOICE
다수의 Primitive Type 또는 중첩된 SEQUENCE 유형을 열거하여 그 중에 한 가지를 선택적으로 표현하는 유형
- SEQUENCE OF / SET OF
단일한 유형을 다수 저장하는 배열형 자료를 표현하는 유형

7.4.1.2. Protocol Data Description

Protocol Data Description은 Protocol Agent가 Protocol을 Simulation 하는데 필요한 Data를 정의한다. Data를 정의하는 방법은 Message Spec을 정의하는 방법을 차용하여 정의한다.

7.4.1.3. Protocol State Description

Protocol이 복잡한 일련의 알고리즘을 따르는 경우, 이를 수용하고 이행할 수단이 필요하다. 따라서 Finite State Machine이 Protocol State Description의 정보를 이용하여 Protocol의 기능을 수행할 수 있도록 하여야 한다.

Protocol State Description은 다음과 같은 내용을 포함하여야 한다.

- 가질 수 있는 상태의 개수
- 받을 수 있는 이벤트의 개수
- (상태-이벤트)에 따른 상태 천이 알고리즘

상태 천이 알고리즘은 Interpreter를 위한 Script 형태로 구성되며 개발을 진행하며 구체적으로 구현하도록 한다.

7.4.1.4. Timing Behavior

대부분의 Protocol은 Response Timeout 및 Retransmission 등과 같은 Timing 동작을 포함하고 있다. 따라서 Protocol Spec에서는 다음과 같은 정보가 포함되어야 한다.

- 타이머의 정의
- 타이머에 따른 동작

타이머의 정의 및 동작은 Interpreter를 위한 Script 형태로 구성되며 개발을 진행하며 구체적으로 구현하도록 한다.

8. 시나리오 엔진 연동시스템

통합관제센터와 교통안전시설 인프라(교통안전 보조시설 포함), 자율차 등 정보연계 대상 간 표준 준용 여부 검증에 필요한 표준검증플랫폼 세부 시스템 및 구현 기능을 설계한다

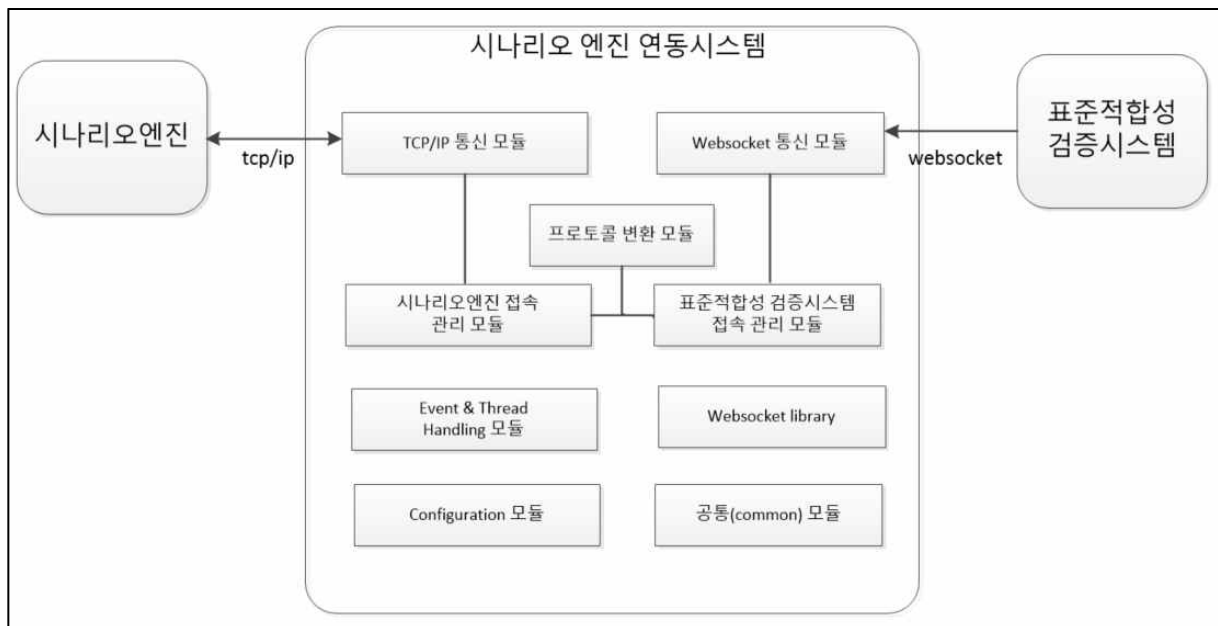
본 문서에서 기술하는 세부 시스템의 요소기술에는 아래 항목이 포함된다

- 자율주행 통합관제센터 시나리오 엔진 연동시스템
- 자율주행 통합관제센터 표준적합성 검증시스템

8.1. S/W Block 설계

8.1.1. S/W Block 구조

아래 그림은 시나리오엔진 연동시스템 내부 구조를 나타내는 S/W Block 구조도이다



(그림 .92) 시나리오엔진 연동시스템 내부구조

8.1.2. 모듈 별 기능 정의

8.1.2.1. tcp/ip 통신 모듈

tcp/ip 통신 모듈은 시나리오엔진 연동시스템이 시나리오엔진과 통신을 하는 모듈이다. 시나리오엔진이 초기 구동 시 자신을 시나리오엔진 연동시스템에 등록하여 자신을 알리는데 이러한 과정에서 사용하는 통신 프로토콜을 구현한 모듈이다

tcp/ip 통신 프로토콜의 자세한 사양(specification)은 추후 별도의 문서에서 정의한다.

8.1.2.2. 시나리오 엔진 접속 관리 모듈

시나리오엔진 접속 관리 모듈은 시나리오엔진의 구동 및 접속 상태를 관리하는 모듈로 표준적합성 검증시스템에서 검증 요청이 수신될 경우 해당 모듈에 등록된 시나리오엔진이 없을 경우 에러를 반환하도록 한다. 또한 다수의 시나리오엔진이 동시 접속될 수 있는 구조로 모듈을 설계하여 다수의 표준적

합성 검증을 동시에 진행할 수 있도록 한다.

8.1.2.3. Websocket 통신 모듈

Websocket 통신 모듈은 표준적합성 검증시스템과 통신을 하기 위한 모듈로 양방향(bi-directional) 통신이 가능해야 하므로 websocket 기술을 이용하여 상호 간의 data 교환이 가능하게 한다

8.1.2.4. 표준적합성 검증시스템 접속 관리 모듈

시나리오엔진 연동시스템이 표준적합성 검증시스템에 연결될 때 이에 대한 접속 정보를 관리하며, 표준적합성 검증시스템과의 접속이 되지 않은 경우에는(혹은 끊어진 경우) 추후 시나리오엔진의 Report event(검증결과) 패킷 전달 시 해당 오류를 시나리오엔진에게 전달한다

8.1.2.5. 프로토콜 변환 모듈

프로토콜 변환 모듈은 사용자의 검증 요청을 시나리오엔진 연동시스템에게 전달하는 과정에서 발생하는 프로토콜을 변환하는 모듈이며, 시나리오엔진 연동시스템에서 발생하는 결과를 검증 요청을 한 사용자에게 전달하는 과정에도 해당 모듈이 프로토콜 변환에 관여한다.

8.1.2.6. Event & Thread handling 모듈

시나리오엔진 연동시스템이 내부적으로 사용하는 thread 와 socket 이벤트 등을 관리하는 모듈이다

8.1.2.7. Configuration 모듈

System configuration에 대한 처리(읽기 및 쓰기)를 담당하는 모듈이다.

8.1.2.8. Common 공통 모듈

Logging 그리고 Helper class와 기타 library를 가지고 있는 모듈이다.

8.1.3. SW 개발 환경 및 사용 기술

8.1.3.1. SW 개발 환경

- 개발 OS

Ubuntu 16 server

- 개발 컴파일러 및 언어

GNU gcc v5.4

- Websocket library

websocketpp (<https://github.com/zaphoyd/websocketpp>)

8.1.3.2. SW 사용 기술

- TCP/IP

시나리오엔진 시스템과 양방향 통신을 위해 tcp/ip 통신을 사용한다

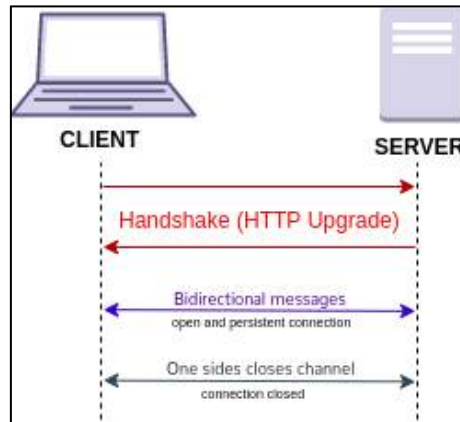
통신 시 사용하는 자세한 프로토콜 규약은 추후 별도 문서에서 정의한다

- Websocket

표준적합성 검증시스템과 양방향 통신을 위해 websocket 기술을 사용한다

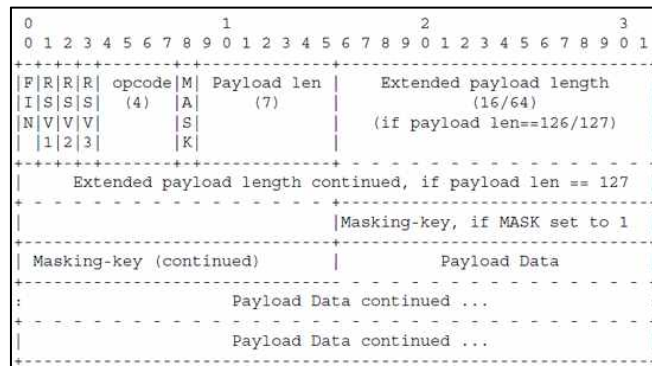
Websocket은 http를 통해 웹브라우저(혹은 일반 client)와 웹서버가 양방향으로 통신할 수 있는 기술이며 이때 초기 연결은 http를 통해 상호 접속을 하며 이후 “HTTP/1.1 101 Switching Protocols” 응답 및 “Upgrade: websocket” 헤더를 통해 websocket 연결을 확립한다.

아래 그림은 웹브라우저(혹은 일반 client)가 웹서버와 http hand-shaking을 통해 websocket 연결을 활성화 하는 그림이다.



(그림.93) websocket 접속 절차

아래 그림은 websocket의 base framing protocol 포맷을 나타낸 그림이다.



(그림.94) websocket 헤더와 페이로드 구조

WebSocket Base framing protocol의 각 항목이 의미하는 바는 다음과 같다

- 1.FIN : 1 bit 뒤에 오는 데이터가 더 있는지 확인하기 위한 패킷
- 2.RSV : 3bit 체크하지 않음
- 3.OPCODE : Frame의 종류 2바이트
- 4.MASK : 1bit, 클라이언트가 서버로 보낼때는 항상 1의 값.
- 5.payload len :크기가 126보다 적으면 payload length가 값을 나타낸다.
- 6.Masking-key : 4 bytes MASK가 1값이면 표시하고 아니면 표시하지 않는다.
- 7.Payload Data : 연결이 설정된 경우, 사용자 지정 확장 데이터가 포함

* WebSocket 접속

wss(TLS/SSL)를 반드시 지원해야 한다.

* WebSocket 버전

RFC 6455 Opening Handshake의 버전은 13을 사용한다. 아래 명시된 헤더는 반드시 포함되어야 하고, 지정된 값이어야 한다. 명시되지 않은 헤더는 모두 무시된다.

- Request URI
 - path : "/"
 - HTTP version : "1.1"
 - Method : "GET"
- Request Headers
 - Host : 서버주소:서버포트
 - Connection : "Upgrade"
 - Sec-WebSocket-Key : nonce 16byte(클라이언트 random, base 64)
 - Origin : 현재 아무값
 - Sec-WebSocket-Version : 13

응답은 아래와 같아야 한다.

- Response Code
 - WWW-Authenticate는 사용하지 않는다.
 - Redirect 하지 않는다.
 - 성공시 101
- Response Headees
 - Connection : "Upgrade"
 - Upgrade : "websocket"
 - Sec-WebSocket-Accept : client nonce 해쉬값

사용 예)

GET / HTTP/1.1

Connection: Upgrade

Upgrade: websocket

Host: 192.168.13.90:3000

Origin: http://www.websocket.org

Sec-WebSocket-Version: 13

Sec-WebSocket-Key: Q22lj2sDqBatsl77BiSBtQ==

HTTP/1.1 101 Switching Protocols

Connection: Upgrade

Upgrade: websocket

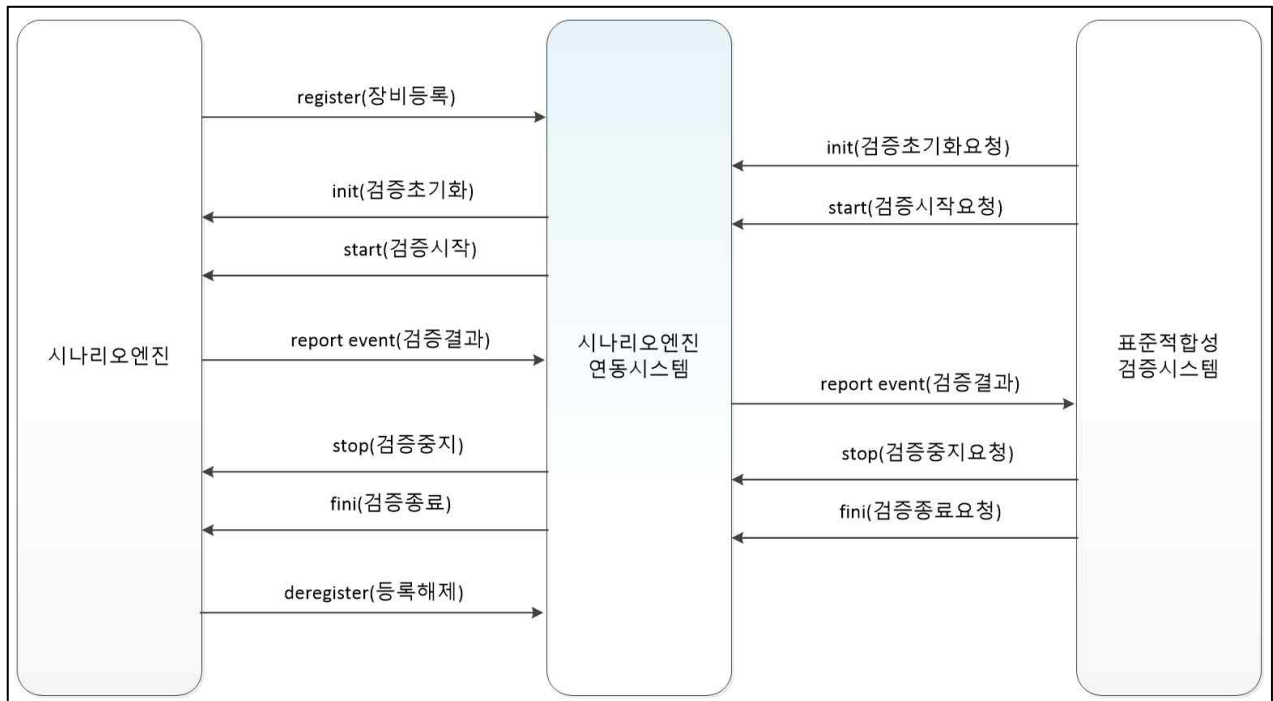
Sec-WebSocket-Accept: Bdx94B9yZLHTGSUFEGtQU3USvf8=

8.1.4. SW 연동 방안

8.1.4.1. 표준적합성 검증시스템과 시나리오엔진 연동 절차

표준적합성 검증시스템과 시나리오엔진 연동시스템은 연동 시 websocket 프로토콜을 이용하여 상호 통신하며 시나리오엔진과 시나리오엔진 연동시스템은 연동 시 tcp/ip 프로토콜을 이용하여 상호 통신한다.

아래 그림은 시나리오엔진 연동시스템이 표준적합성 검증시스템 및 시나리오엔진과 상호 연동하는 방식을 나타낸 그림이다.



(그림 .95) 시나리오엔진과 표준적합성 검증시스템과의 연동 절차

먼저 시나리오엔진은 자신을 시나리오엔진 연동시스템에게 등록(register)하여 표준적합성 검증시스템 사용자가 검증을 시작할 수 있도록 준비한다

이후 표준적합성 검증시스템의 사용자가 웹화면에서 검증 시작을 선택하면 표준적합성 검증시스템은 시나리오엔진 연동시스템에 init(검증초기화요청) 및 start(검증시작요청)을 패킷을 전송한다

init(검증초기화요청) 및 start(검증시작요청)을 패킷을 수신한 시나리오엔진 연동시스템은 해당 패킷을 tcp/ip 프로토콜에 맞게 변환한 후 해당 요청을 시나리오엔진에게 전달한다.

검증시작 요청을 수신한 시나리오엔진은 검증이 완료되면 검증 결과를 포함한 report event(검증결과)를 시나리오엔진 연동시스템에게 전송하며 이는 다시 표준적합성 검증시스템에게 전달되어 표준적합성 검증시스템은 이를 사용자의 웹화면에 표시하게 된다.

검증결과를 수신한 사용자가 웹화면에서 로그아웃하거나 검증 종료를 선택하면 표준적합성 검증시스템은 stop(검증중지요청) 및 fini(검증종료요청) 패킷을 시나리오엔진 연동시스템에 전송하게 되며 시나리오엔진 연동시스템은 이를 시나리오엔진에게 전달한 후 시나리오엔진은 더 이상 자신에게 요청된 검증 요청이 없을 경우 시나리오엔진 연동시스템에게 자신을 deregister(등록해제)하게 된다.

9. 표준적합성 검증시스템

통합관제센터와 교통안전시설 인프라(교통안전 보조시설 포함), 자율차 등 정보연계 대상 간 표준 준용 여부 검증에 필요한 표준검증플랫폼 세부 시스템 및 구현 기능을 설계한다

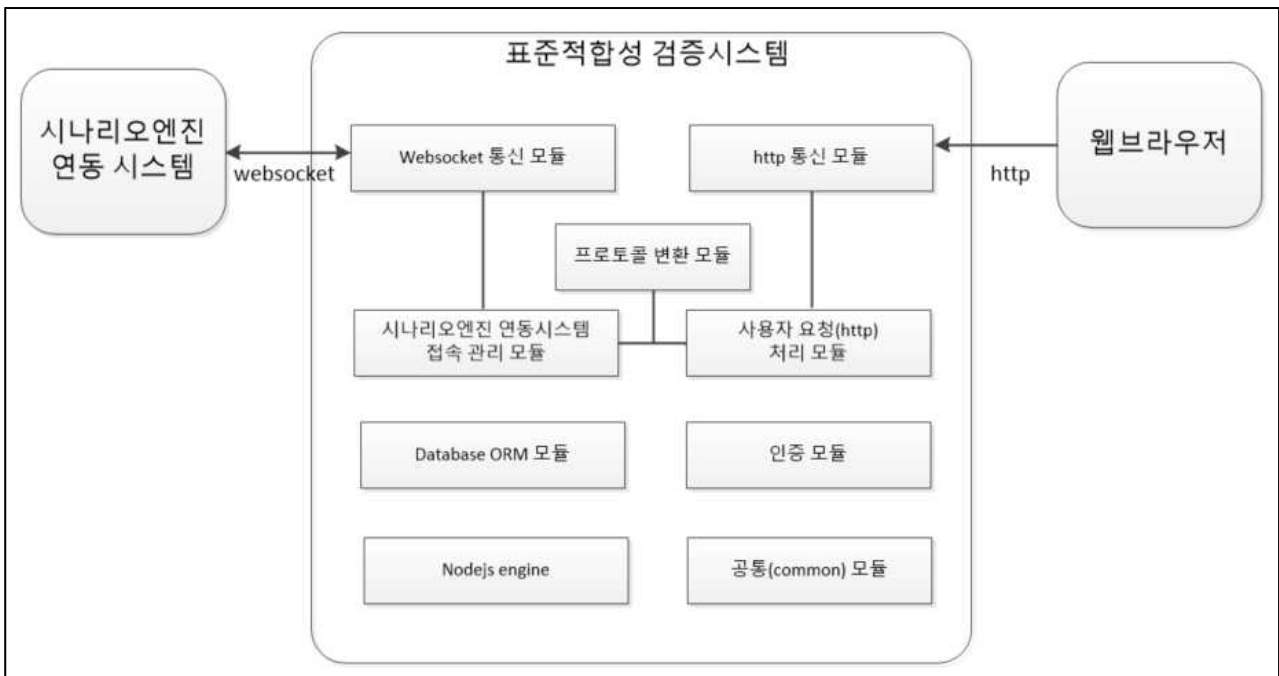
본 문서에서 기술하는 세부 시스템의 요소기술에는 아래 항목이 포함된다

- 자율주행 통합관제센터 표준적합성 검증시스템
- 자율주행 통합관제센터 시나리오 엔진 연동시스템

9.1. S/W Block 설계

9.1.1. S/W Block 구조

아래 그림은 표준적합성 검증시스템 내부 구조를 나타내는 S/W Block 구조도이다



(그림 .96) 표준적합성 검증시스템의 내부 구조

9.1.2. 모듈 별 기능 정의

9.1.2.1. http 통신(사용자 요청 처리) 모듈

http 통신 모듈은 사용자가 표준적합성 검증을 신청 시 해당 http 요청을 수신하고 이에 대한 처리를 다른 모듈에게 위임하는 모듈이다. 사용자 요청 처리 모듈은 다수의 http 요청을 동시에 처리해야 하므로 node.js 엔진을 이용하여 구현한다

9.1.2.2. 인증 모듈

인증 모듈은 Web 인터페이스를 통해 표준적합성 검증 신청 시 사용자를 식별 및 인증하고 해당 사용자가 검증 신청을 할 수 있는 사용자 권한이 존재하는지 여부를 검사하는 모듈이다.

9.1.2.3. Websocket 통신 모듈

Websocket 통신 모듈은 시나리오엔진 연동시스템과 통신을 하기 위한 모듈로 양방향 (bi-directional) 통신이 가능해야 하므로 websocket 기술을 이용하여 상호 간의 data 교환이 가능하게 한다

9.1.2.4. 시나리오엔진 연동시스템 접속 관리 모듈

시나리오엔진 연동시스템 접속 관리 모듈은 시나리오엔진 연동시스템이 표준적합성 검증시스템에 연결될 때 이에 대한 접속 여부를 관리하며, 시나리오엔진 연동시스템이 접속이 되지 않은 경우에는 추후 사용자의 검증 신청 시 오류를 표시한다.

9.1.2.5. 프로토콜 변환 모듈

프로토콜 변환 모듈은 사용자의 검증 요청을 시나리오엔진 연동시스템에게 전달하는 과정에서 발생하는 프로토콜을 변환하는 모듈이며, 시나리오엔진 연동시스템에서 발생하는 결과를 검증 요청을 한 사용자에게 전달하는 과정에도 해당 모듈이 프로토콜 변환에 관여한다.

9.1.2.6. Database ORM 모듈

시나리오엔진 연동시스템과 사용자의 정보 등 임시 보관 용이 아닌 영구 보관 성격의 데이터를 보관 하기 위해 사용하는 모듈이다

9.1.2.7. Common 공통 모듈

System configuration과 logging 그리고 Helper class와 기타 library를 가지고 있는 모듈이다.

9.1.3. SW 개발 환경 및 사용 기술

9.1.3.1. SW 개발 환경

- 개발 OS

Ubuntu 16 server

- 개발 플랫폼 및 언어

Nodejs v10.19.0 / Javascript

- 개발 Database

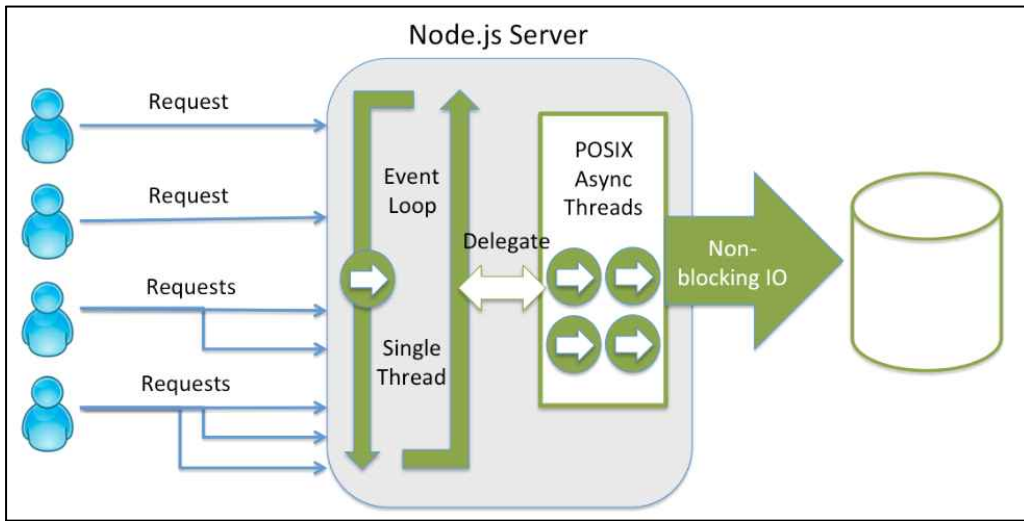
SQLite

9.1.3.2. SW 사용 기술

- Nodejs

표준적합성 검증시스템은 http를 통해 사용자로부터 검증 신청을 수신하고 이를 시나리오엔진 연동 시스템과 연동하여 해당 결과를 사용자에게 전달한다. 이때 다수의 http 요청이 동시 처리 가능해야 하며 추후 기술될 websocket 연동 또한 가능해야 하기 때문에 이러한 요구사항을 만족할 수 있는 웹 플랫폼 중 하나인 nodejs를 사용해서 표준적합성 검증시스템을 구현한다

아래 그림은 nodejs가 사용자의 요청을 처리하는 방식을 설명한 그림이다.



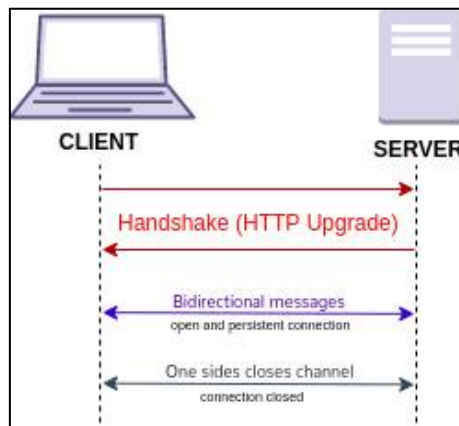
(그림 .97) nodejs 의 내부 구조

○ Websocket

시나리오엔진 연동시스템과 양방향 통신을 위해 websocket 기술을 사용한다

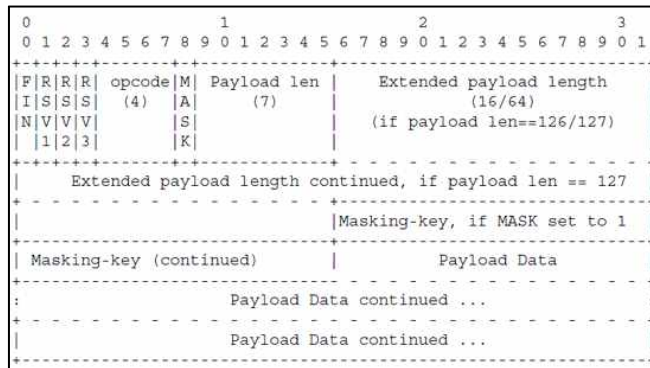
Websocket은 http를 통해 웹브라우저(혹은 일반 client)와 웹서버가 양방향으로 통신할 수 있는 기술이며 이때 초기 연결은 http를 통해 상호 접속을 하며 이후 “HTTP/1.1 101 Switching Protocols” 응답 및 “Upgrade: websocket” 헤더를 통해 websocket 연결을 확립한다.

아래 그림은 웹브라우저(혹은 일반 client)가 웹서버와 http hand-shaking을 통해 websocket 연결을 활성화 하는 그림이다.



(그림 .98) websocket 접속 절차

아래 그림은 websocket의 base framing protocol 포맷을 나타낸 그림이다.



(그림.99) websocket 헤더와 페이로드 구조

Websocket Base framing protocol의 각 항목이 의미하는 바는 다음과 같다

- 1.FIN : 1 bit 뒤에 오는 데이터가 더 있는지 확인하기 위한 패킷
- 2.RSV : 3bit 체크하지 않음
- 3.OPCODE : Frame의 종류 2바이트
- 4.MASK : 1bit, 클라이언트가 서버로 보낼때는 항상 1의 값.
- 5.payload len :크기가 126보다 적으면 payload length가 값을 나타낸다.
- 6.Masking-key : 4 bytes MASK가 1값이면 표시하고 아니면 표시하지 않는다.
- 7.Payload Data : 연결이 설정된 경우, 사용자 지정 확장 데이터가 포함

* WebSocket 접속

wss(TLS/SSL)를 반드시 지원해야 한다.

* WebSocket 버전

RFC 6455 Opening Handshake의 버전은 13을 사용한다. 아래 명시된 헤더는 반드시 포함되어야 하고, 지정된 값이어야 한다. 명시되지 않은 헤더는 모두 무시된다.

- Request URI
 - path : "/"
 - HTTP version : "1.1"
 - Method : "GET"
- Request Headers
 - Host : 서버주소:서버포트
 - Connection : "Upgrade"
 - Sec-WebSocket-Key : nonce 16byte(클라이언트 random, base 64)
 - Origin : 현재 아무값
 - Sec-WebSocket-Version : 13

응답은 아래와 같아야 한다.

- Response Code
 - WWW-Authenticate는 사용하지 않는다.
 - Redirect 하지 않는다.
 - 성공시 101
- Response Headees
 - Connection : "Upgrade"
 - Upgrade : "websocket"
 - Sec-WebSocket-Accept : client nonce 해쉬값

HTTP를 통한 WebSocket 접속 예)

GET / HTTP/1.1

Connection: Upgrade

Upgrade: websocket

Host: 192.168.13.90:3000

Origin: http://www.websocket.org

Sec-WebSocket-Version: 13

Sec-WebSocket-Key: Q22lj2sDqBatsl77BiSBtQ==

HTTP/1.1 101 Switching Protocols

Connection: Upgrade

Upgrade: websocket

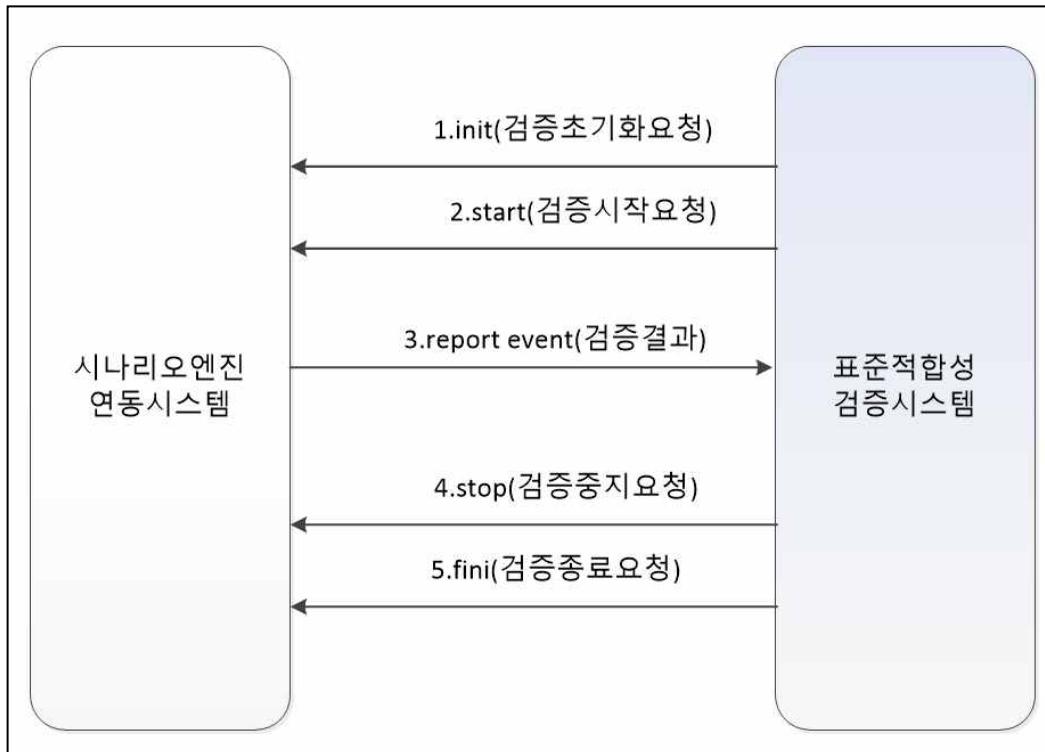
Sec-WebSocket-Accept: Bdx94B9yZLHTGSUFEGtQU3USvf8=

9.1.4. SW 연동 방안

9.1.4.1. 시나리오엔진 연동시스템 연동 절차

앞서 언급한 바와 같이 표준적합성 검증시스템과 시나리오엔진 연동시스템은 연동 시 websocket 프로토콜을 이용하여 상호 통신한다.

아래 그림은 시나리오엔진 연동시스템이 표준적합성 검증시스템과 상호 연동하는 방식을 나타낸 그림이다.



(그림.100) 표준적합성 검증시스템과 시나리오엔진 연동시스템의 연동 절차

표준적합성 검증시스템의 사용자가 웹화면에서 검증 시작을 선택하면 표준적합성 검증시스템은 시나리오엔진 연동시스템에 init(검증초기화요청) 및 start(검증시작요청)을 패킷을 전송한다

검증시작요청(start)을 수신한 시나리오엔진 시스템은 검증이 완료되면 검증 결과를 포함한 report event(검증결과)를 표준적합성 검증시스템에게 전송하며 표준적합성 검증시스템은 이를 사용자의 웹화면에 표시하게 된다.

검증결과를 수신한 사용자가 웹화면에서 로그아웃하거나 검증 종료를 선택하면 표준적합성 검증시스템은 stop(검증중지요청) 및 fini(검증종료요청) 패킷을 시나리오엔진 연동시스템에 전송하게 된다.

10. 사용자 매뉴얼

10.1. 표준검증플랫폼의 기능

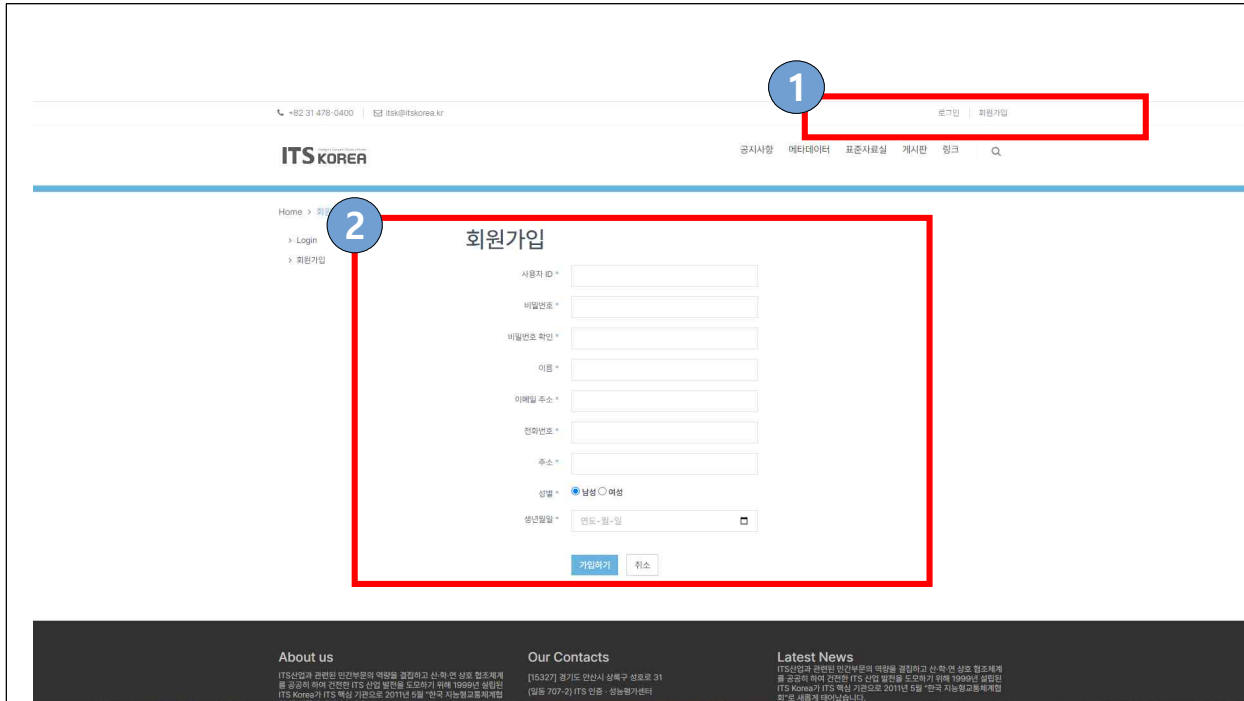
- 자율주행 통합교통관제센터 표준검증플랫폼은 통합관제센터와 교통안전시설 인프라(교통 안전 보조시설 포함), 자율주행차 등 정보연계 대상 간 표준 준용 여부 검증에 필요한 표준검증플랫폼 세부 시스템 및 구현을 목표로 개발되었으며, 통합관제시스템에 연동되는 시스템들의 표준 준용을 통해 전체 플랫폼의 안정성을 보장하고, 표준 적합성 테스트 환경을 구축, 범용적으로 표준적합성 테스트를 수행할 수 있는 기능을 제공한다. 본 문서에서 설명하는 세부 시스템에는 아래 항목이 포함된다.

- 자율주행 통합관제센터 메타데이터 관리시스템

- 본 사용자 매뉴얼은 표준검증플랫폼 서비스를 이용하기 위한 절차 및 방법을 기술한다. 표준검증플랫폼 서비스를 처음 접하는 사용자를 위해 사용 방법을 요약, 안내하는 데 작성 목적이 있다. 사용자 매뉴얼의 저작권은 개발사에 있으며, 개발사의 허락 없이 임의 수정을 할 수 없다.

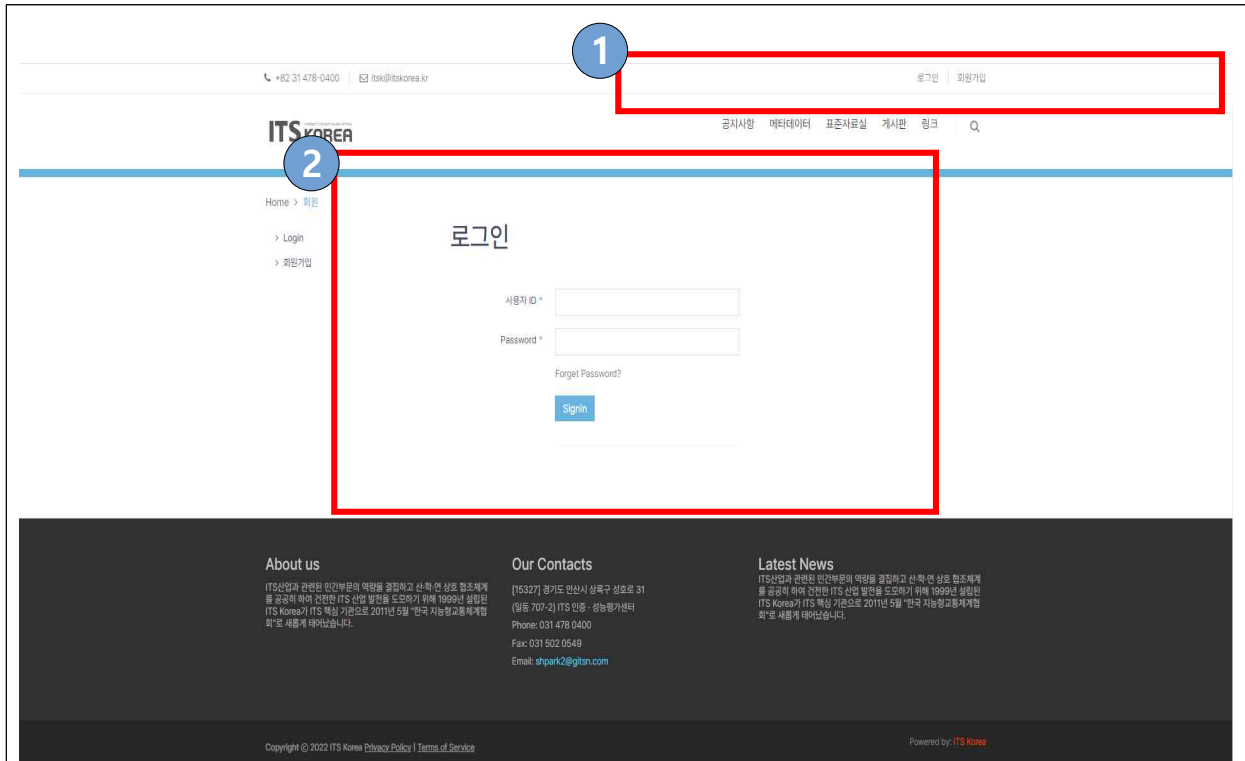
10.2. 기능별 사용방법

10.2.1. 회원가입



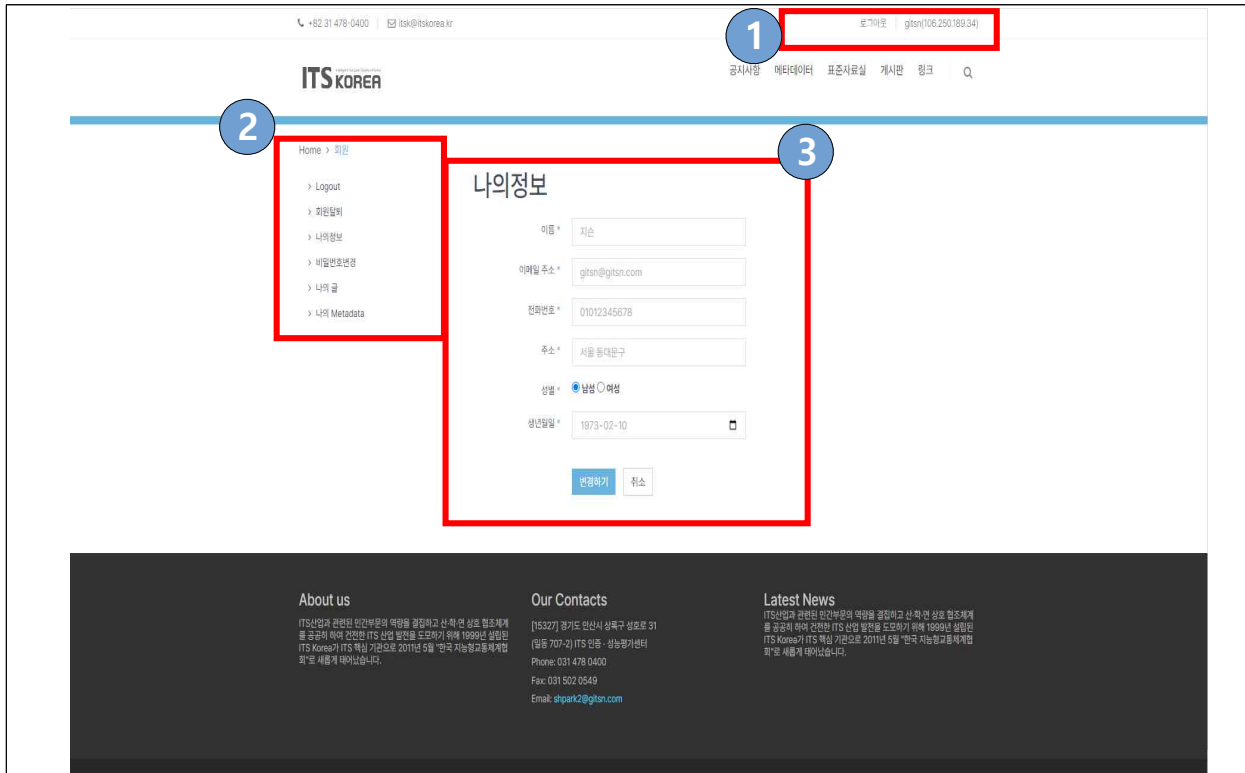
화면 개요	회원가입 화면
화면 설명	1 상단에 표시되는 우측 상단 회원가입 링크를 이용 - 클릭 시 회원가입 화면으로 이동
	2 가입 형식에 맞게 기재 후 '가입하기' 버튼 클릭
사용 방법	<ol style="list-style-type: none"> 1. 메뉴 : 최상단 '회원가입' 링크 > 회원가입 화면 2. 초기화면 - 회원가입 양식 화면 표출 3. 선택화면 - 가입 후 관리자가 승인 시 로그인 가능
특기 사항	없음

10.2.2. 로그인



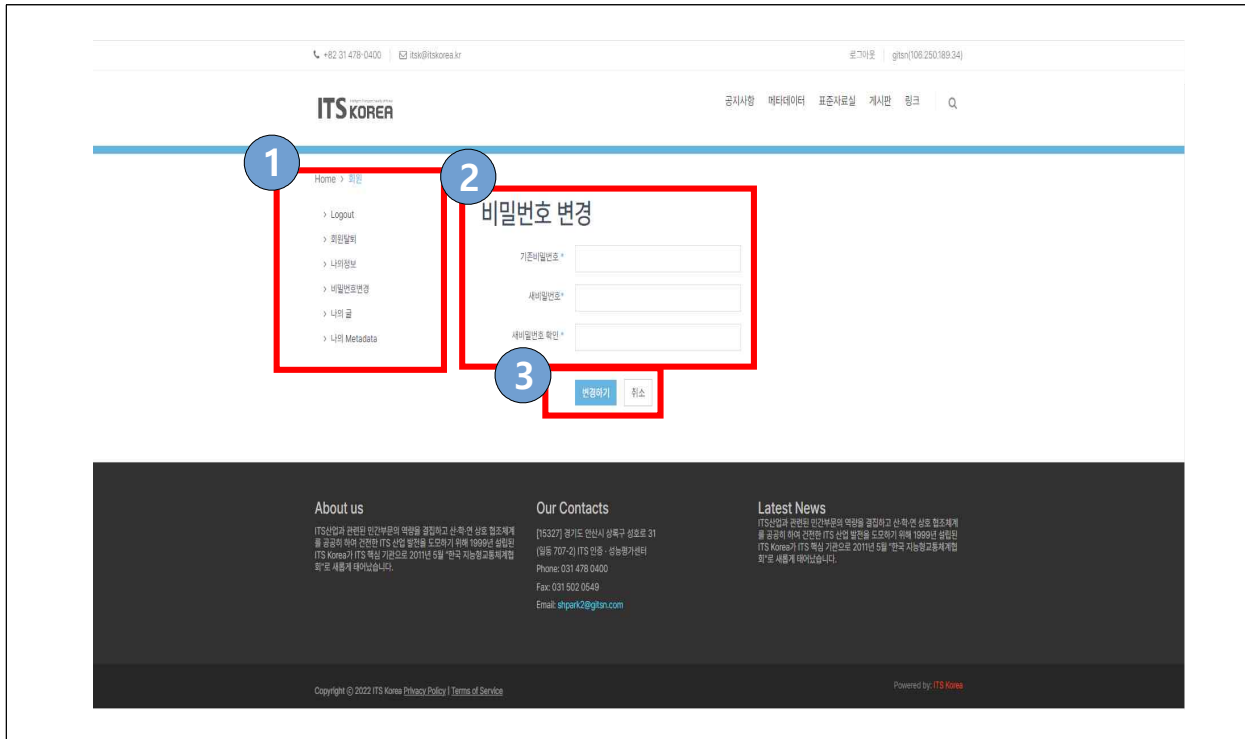
화면 개요	로그인 화면	
화면 설명	1	로그인으로 갈 수 있는 링크
	2	로그인 화면
사용 방법	1. 메뉴 : 우측 상단 로그인 링크 > 로그인 화면 2. 초기화면 - 로그인 기능 3. 선택화면 - 왼쪽 메뉴를 통해 로그인과 회원가입을 선택할 수 있음	
특기 사항	없음	

10.2.3. 마이 페이지



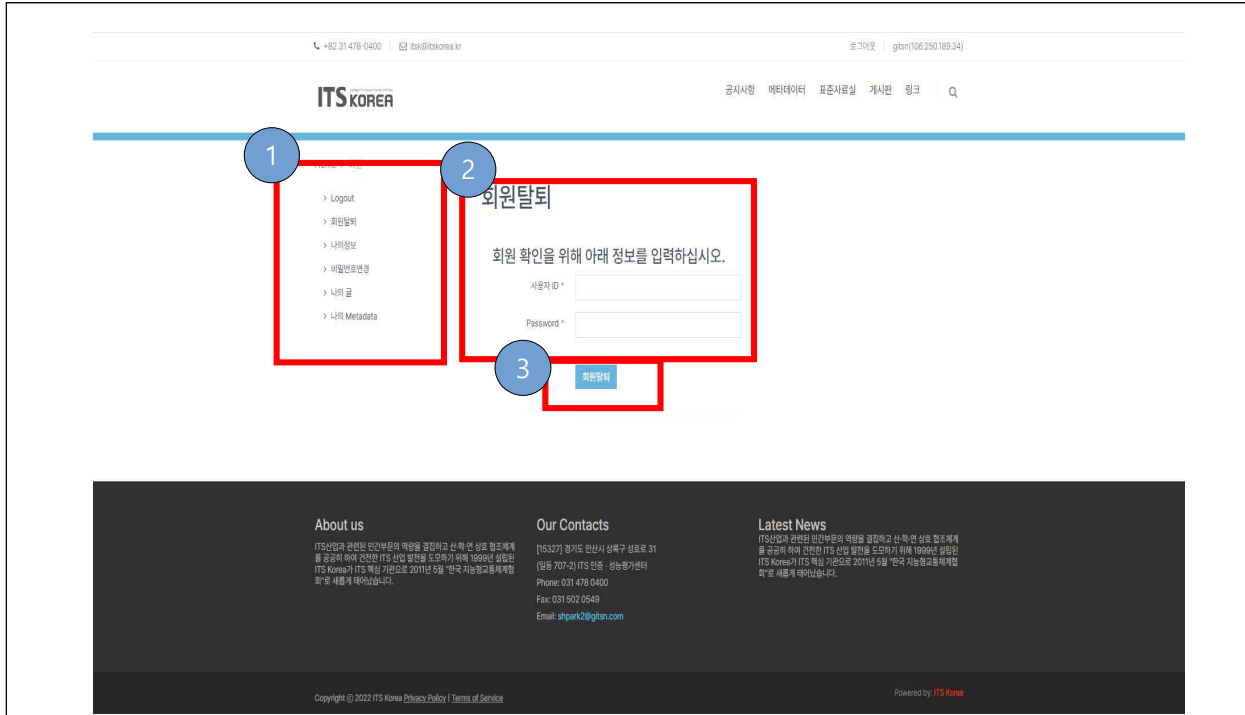
화면 개요	마이페이지 화면 [다양한 서비스로 이루어져 있음]	
화면 설명	1	로그인 시 회원가입 버튼은 사용자의 정보로 갈 수 있는 버튼으로 바뀜
	2	마이페이지의 메뉴들로 로그아웃, 회원탈퇴, 나의 정보 변경, 비밀번호 변경, 나의 글, 나의 Metadata를 선택할 수 있음
	3	우측 상단의 사용자 클릭 시 제공되는 나의 정보 변경 화면이며, 정보를 변경하고 '변경하기' 버튼 클릭을 통해 내용 변경이 가능
사용 방법	1. 메뉴 : 우측 상단 사용자 정보 > 나의 정보 변경 2. 초기화면 - 나의 정보 변경 설정 기능 3. 선택화면 - 왼쪽 메뉴를 통해 원하는 항목을 골라 내용을 변경할 수 있음	
특기 사항	없음	

10.2.4. 비밀번호 변경



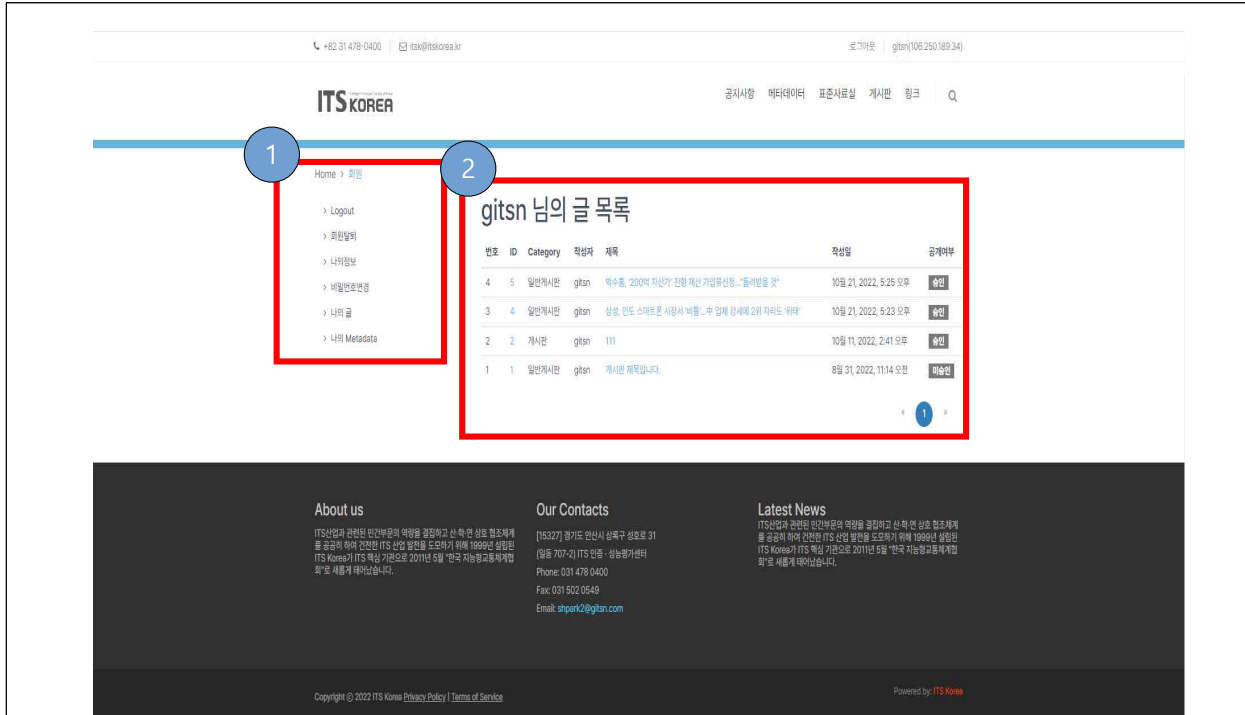
화면 개요	비밀번호 변경 화면	
화면 설명	1	왼쪽 메뉴를 통해 비밀번호 변경 화면으로 이동
	2	기존 비밀번호 및 변경할 새 비밀번호 입력
	3	'변경하기' 버튼 클릭으로 비밀번호 변경 가능
사용 방법	1. 메뉴 : 왼쪽 메뉴 > 비밀번호 변경	
	2. 초기화면 - 기존, 새 비밀번호 입력 란 - '변경하기' 버튼 클릭 후 비밀번호 변경	
특기 사항	없음	

10.2.5. 회원탈퇴



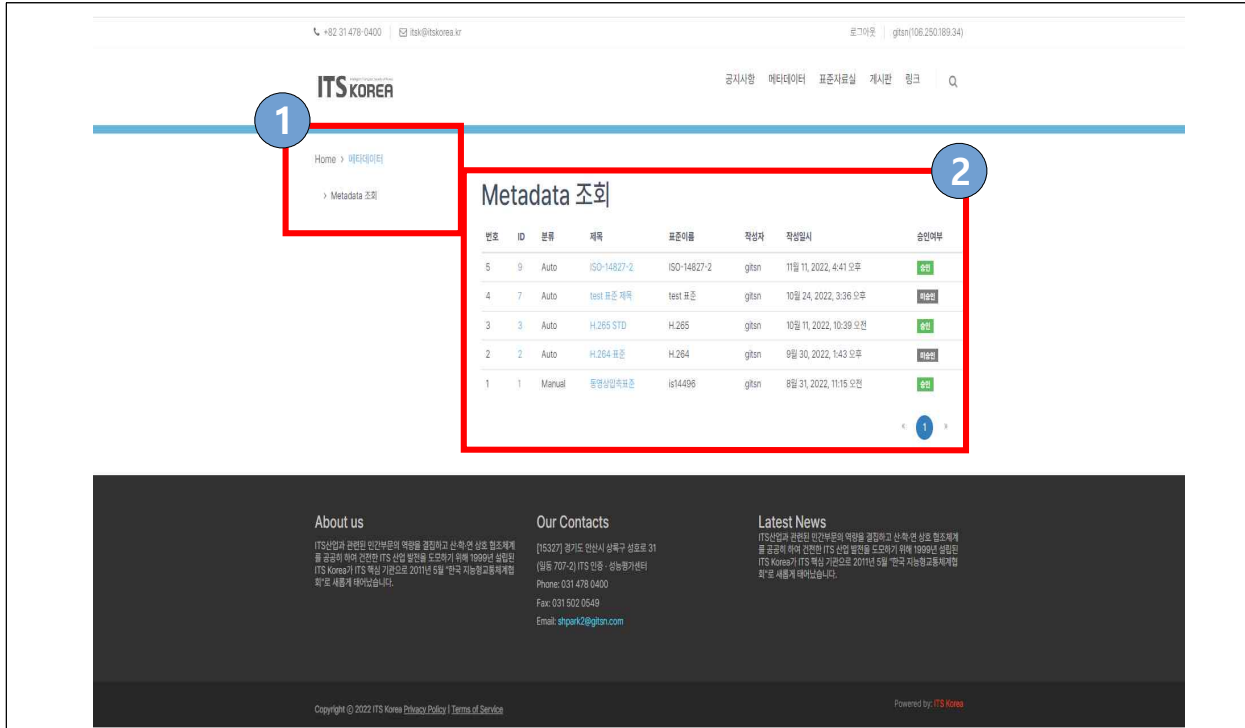
화면 개요	회원탈퇴 화면	
화면 설명	1	왼쪽 메뉴를 통해 회원탈퇴 화면으로 이동
	2	사용자 ID 와 비밀번호 입력 란
	3	'회원탈퇴' 버튼
사용 방법	1. 메뉴 : 왼쪽 메뉴 > 회원탈퇴	
	2. 초기화면 - 사용자 ID, 비밀번호 입력 후 '회원탈퇴' 버튼 클릭 후 회원탈퇴	
특기 사항	3. 선택화면 - 회원 탈퇴 후 메인 화면으로 이동	
	없음	

10.2.6. 나의 글



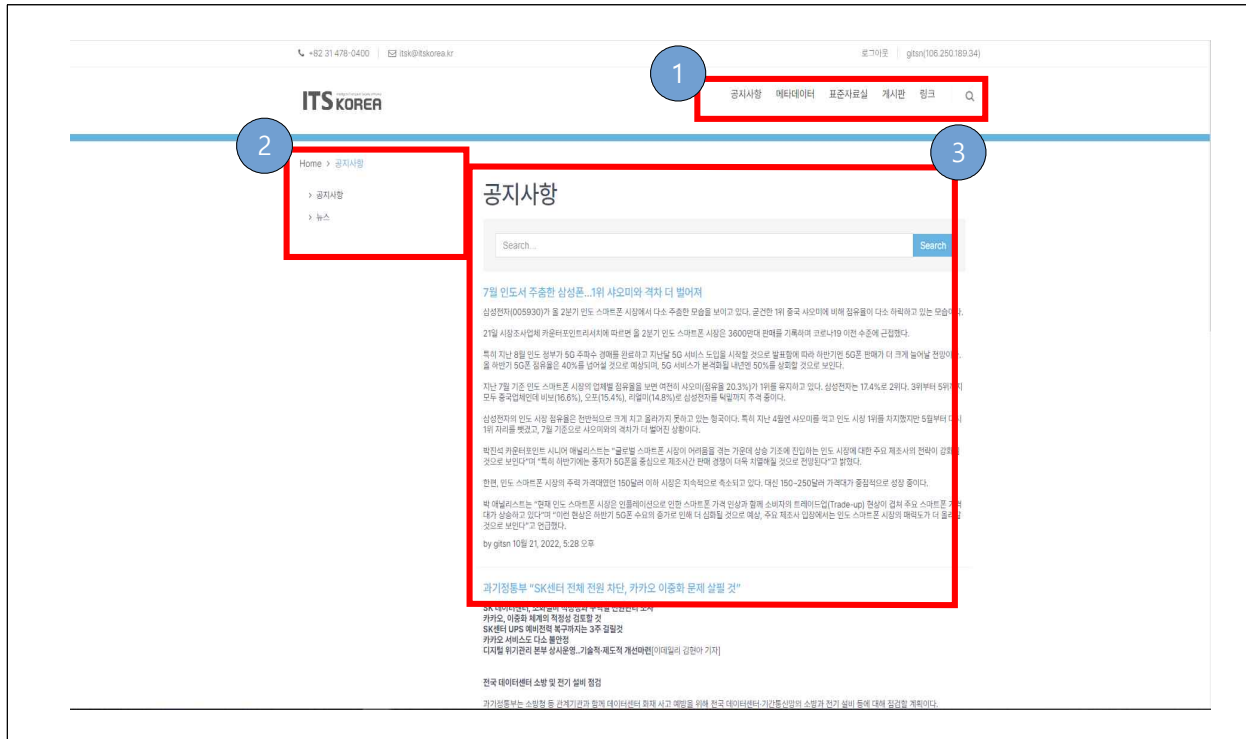
화면 개요	나의 글 화면
화면 설명	1. 왼쪽 메뉴를 통해 나의 글 화면으로 이동
	2. 내가 작성한 글 목록 표시
사용 방법	1. 메뉴 : 왼쪽 메뉴 > 나의 글
	2. 초기화면 - 내가 올린 게시판 글 확인 가능, 공개 여부도 표시
특기 사항	3. 선택화면 - 목록에 있는 글을 클릭하여 내용 확인 가능
	없음

10.2.7. 나의 메타데이터



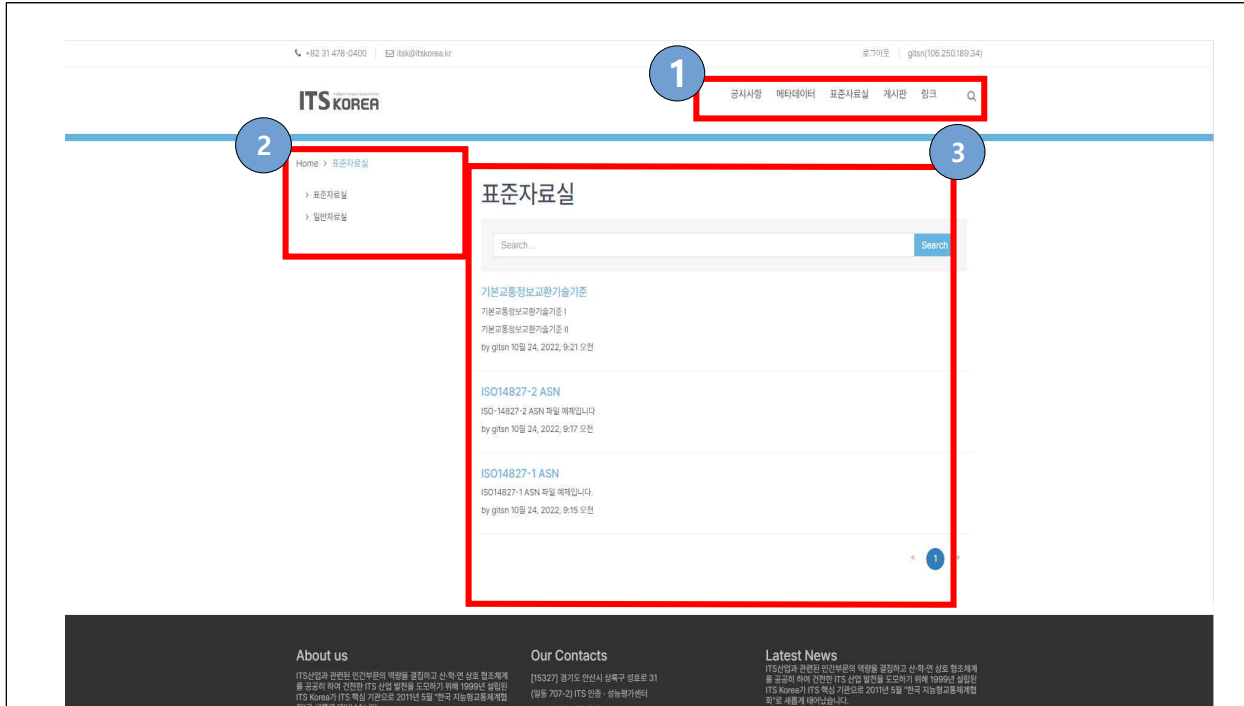
화면 개요	나의 Metadata 화면
화면 설명	1. 왼쪽 메뉴를 통해 나의 Metadata 페이지 이동
	2. 내가 작성한 메타데이터 등록 요청 목록
사용 방법	1. 메뉴 : 왼쪽 메뉴 > 나의 Metadata
	2. 초기화면 - 내가 등록 요청한 메타데이터 확인 가능, 승인 여부도 포함
특기 사항	3. 선택화면 - 목록에 있는 글을 클릭하여 내용 확인 가능
	없음

10.2.8. 공지사항



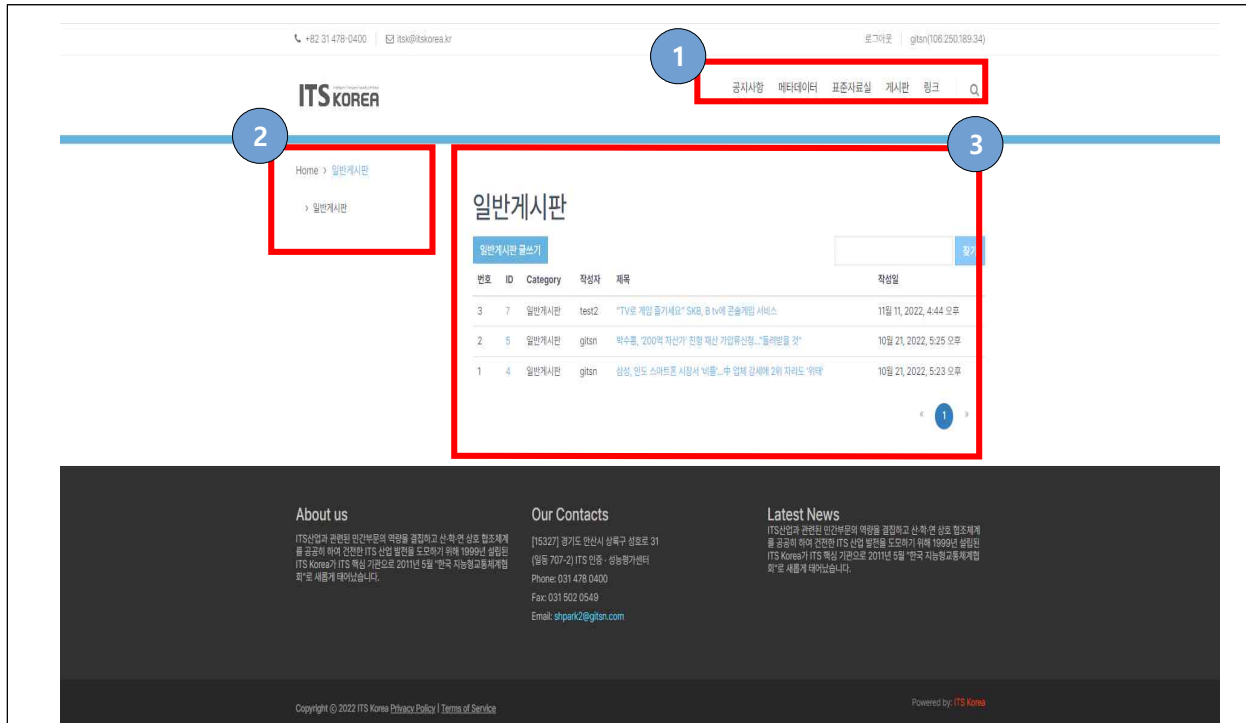
화면 개요	공지사항 목록 조회 화면	
화면 설명	1	메인 메뉴에서 공지사항으로 이동
	2	왼쪽 메뉴 [공지사항, 뉴스]
	3	공지사항 글 목록
사용 방법	1. 메뉴 : 주메뉴 > 공지사항	
	2. 초기화면	- 공지사항 제목, 내용, 작성자, 작성일 표시
특기 사항	3. 선택화면	- 목록에 있는 글을 클릭하여 내용 확인 가능
		뉴스, 링크와 사용법 동일

10.2.9. 자료실



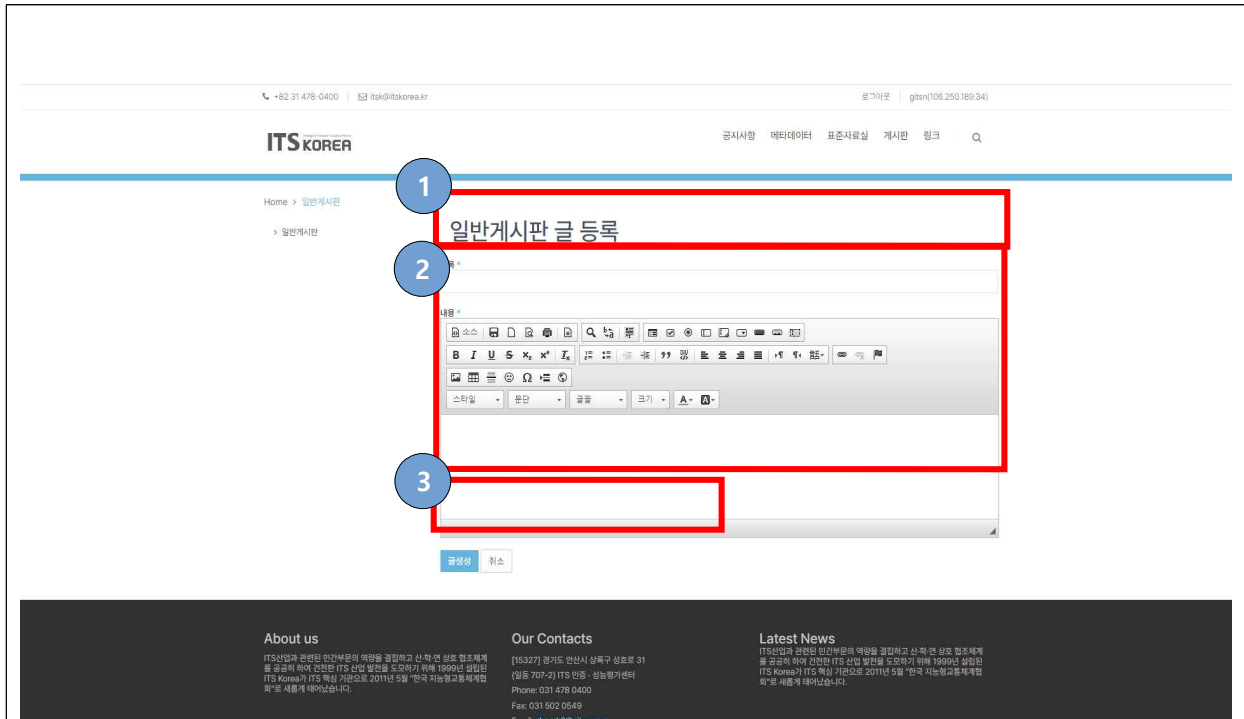
화면 개요	자료실 화면
화면 설명	1 메인 메뉴에서 표준자료실로 이동
	2 왼쪽 메뉴 [표준자료실, 일반자료실]
	3 자료실 글 목록
사용 방법	1. 메뉴 : 주메뉴 > 표준자료실
	2. 초기화면 - 등록된 자료의 제목, 내용, 작성자, 작성일 표시
특기 사항	3. 선택화면 - 목록에 있는 글을 클릭하여 내용 확인 가능
	일반자료실과 사용법 같음

10.2.10. 게시판



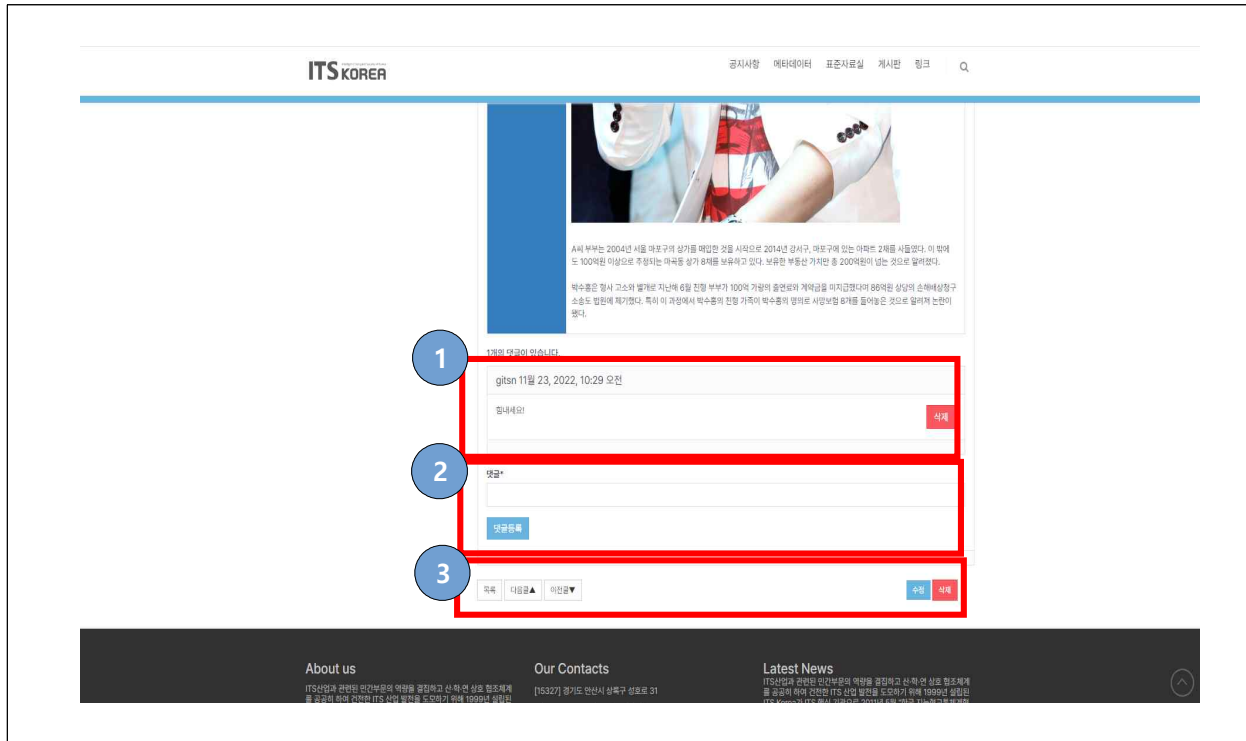
화면 개요	사용자가 게시 가능한 일반 게시판 화면	
화면 설명	1	메인 메뉴에서 게시판으로 이동
	2	왼쪽 메뉴 활성화
	3	게시판 등록 및 목록을 제공
	4	
사용 방법	1. 메뉴 : 게시판 > 일반게시판	
	2. 초기화면 - 일반게시판 항목 목록 제공	
특기 사항	3. 선택화면 - 일반게시글 정보 조회	
	관리자 글 승인 후 사용자 글이 목록에 표시	

10.2.11. 게시판 글 등록



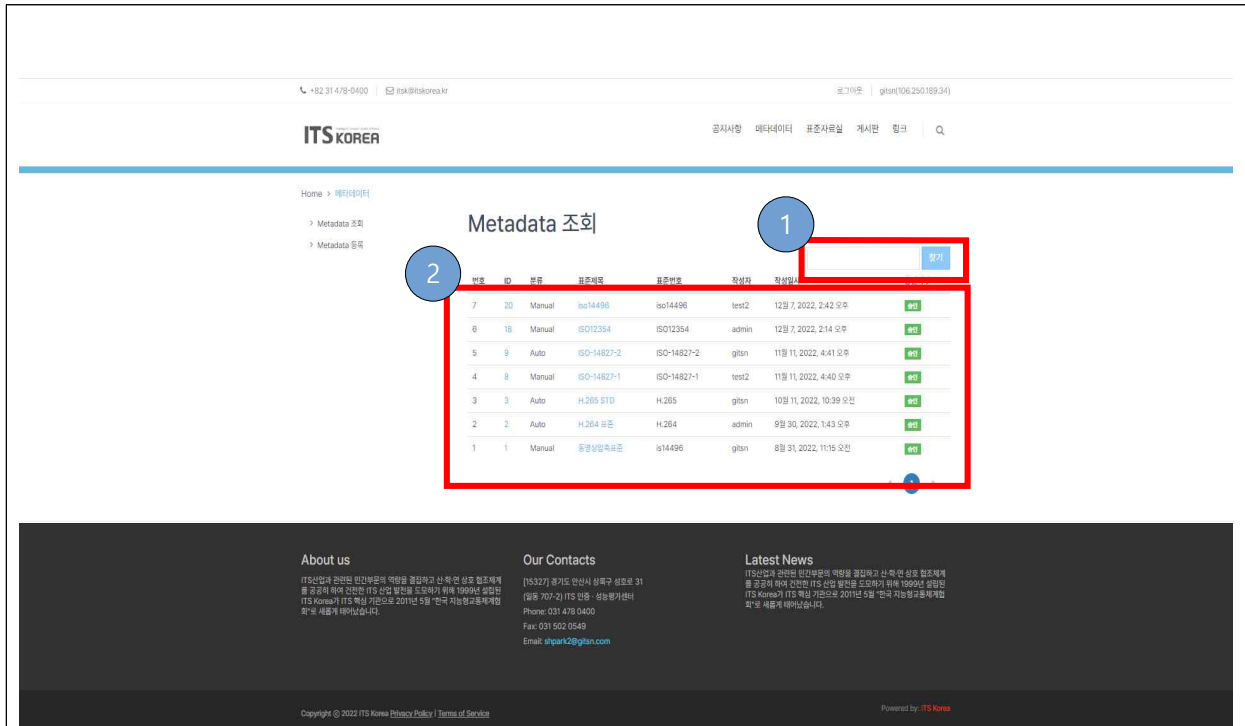
화면 개요	일반게시판 글 등록 화면	
화면 설명	1	제목 기입란
	2	웹편집기를 통해 내용 삽입 가능
	3	'글생성' 버튼 클릭으로 글 생성 가능
사용 방법	1. 메뉴: 일반게시판 목록 > '글생성' 버튼 2. 초기화면 - 제목 및 내용 삽입 후 '글생성' 버튼 클릭하여 글 생성 3. 선택화면 - 게시판 목록에서 나의 글 확인	
특기 사항	사용자가 직접 글 등록 가능, 관리자 글 승인 후 사용자 글이 목록에 표시	

10.2.12. 게시판 댓글



화면 개요	일반게시판 글 댓글 등록 화면	
화면 설명	1	다른 사용자의 댓글 확인 가능
	2	댓글 등록란
	3	목록, 이전, 다음 글 및 게시 글 수정, 삭제 버튼
사용 방법	1. 메뉴: 일반게시판 글 > 게시 글의 하단	
	2. 초기화면 - 댓글 내용을 확인할 수 있고 사용자가 자신의 댓글을 삭제 가능 - 내용 삽입 후 댓글 등록 버튼을 이용해 댓글 등록 가능	
특기 사항	3. 선택화면 - 목록, 이전, 다음 글 클릭을 통해 화면 이동이 가능하며, 수정, 삭제 버튼을 통해 자신의 글을 수정하고 삭제할 수 있음	
	사용자가 다른 사람의 글에 댓글을 등록 가능	

10.2.13. 메타데이터 목록 조회



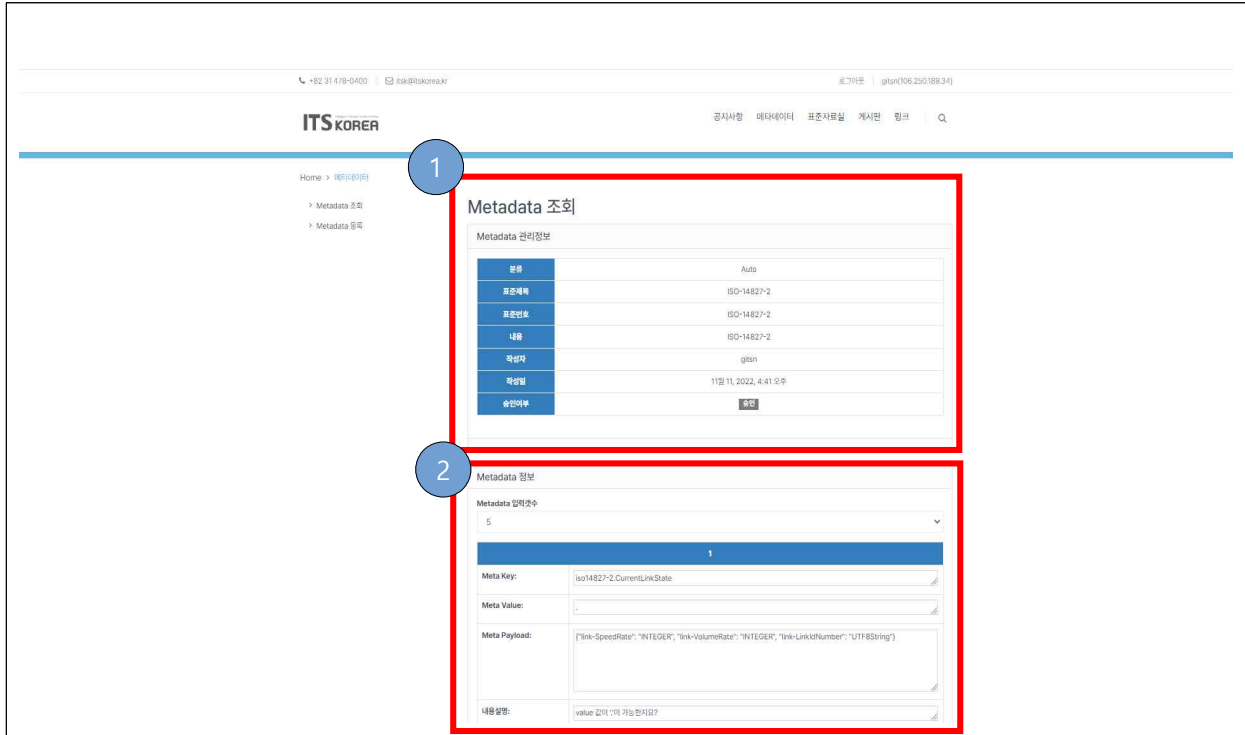
화면 개요	메타데이터 목록 조회 화면
화면 설명	1 키워드 검색 후 찾기 기능
	2 메타데이터 목록 화면
사용 방법	<ol style="list-style-type: none"> 초기화면 <ul style="list-style-type: none"> 키워드 검색 후 키워드가 들어가 있는 메타데이터 목록들만 표시 선택화면 <ul style="list-style-type: none"> 메타데이터의 정보 조회
특기 사항	없음

10.2.14. 메타데이터 신청

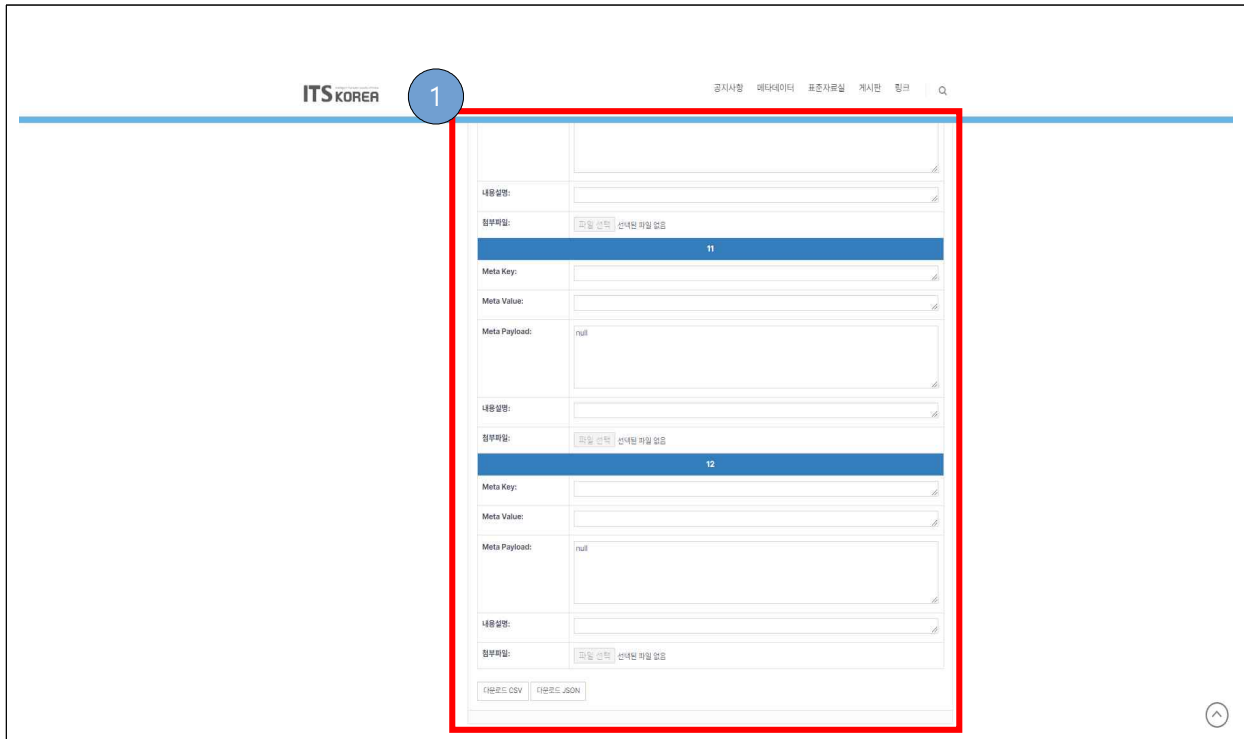


화면 개요	메타데이터 등록 신청 화면	
화면 설명	1	메인 메뉴에서 메타데이터 등록으로 이동
	2	왼쪽 메뉴 활성화
	3	메타데이터 등록 화면
사용 방법	1. 메뉴: 메타데이터 등록 선택	
	2. 초기화면 - 표준제목, 표준번호, 표준내용 입력 화면	
특기 사항	3. 선택화면 - '등록하기' 버튼을 통해 메타데이터 관리정보 등록 가능	
	메타데이터 정보 및 자세한 내용 수정은 등록 후에 가능	

10.2.15. 메타데이터 조회/수정/삭제



화면 개요	메타데이터 조회 화면	
화면 설명	1	메타데이터 관리 정보
	2	메타데이터 정보
사용 방법	1. 메뉴 : 메타데이터 조회 > 메타데이터 관리정보	
	2. 초기화면 - 기존에 등록된 정보를 볼 수 있으며 미승인 상태에서는 수정, 삭제 버튼을 클릭할 수 있다	
특기 사항	3. 선택화면 - 수정을 선택하여 메타데이터 관리정보를 수정할 수 있으며, 삭제 버튼을 통해 메타데이터 관리정보를 삭제할 수 있다.	
	미승인 상태이고 자신이 신청한 메타데이터는 수정, 삭제 가능	



화면 개요	메타데이터 조회 화면	
화면 설명	1	메타데이터 정보 조회 및 수정 화면
사용 방법	1. 메뉴: 목록 조회 > 정보 조회 2. 초기화면 - Meta key, value, payload, 내용설명 직접 입력 후 변경사항적용 버튼 선택 - 이미 등록되어 있는 Meta key, value, payload, 내용설명을 조회 3. 선택화면 - 수정사항 적용 및 메타데이터를 CSV, JSON 파일로 다운로드 가능	
특기 사항	하단에 메타데이터 요청 정보와 이력 정보도 확인 가능 승인된 메타데이터는 수정, 삭제 불가능 다른 사람이 등록한 메타데이터는 수정, 삭제 불가능	

11. 운용자 매뉴얼

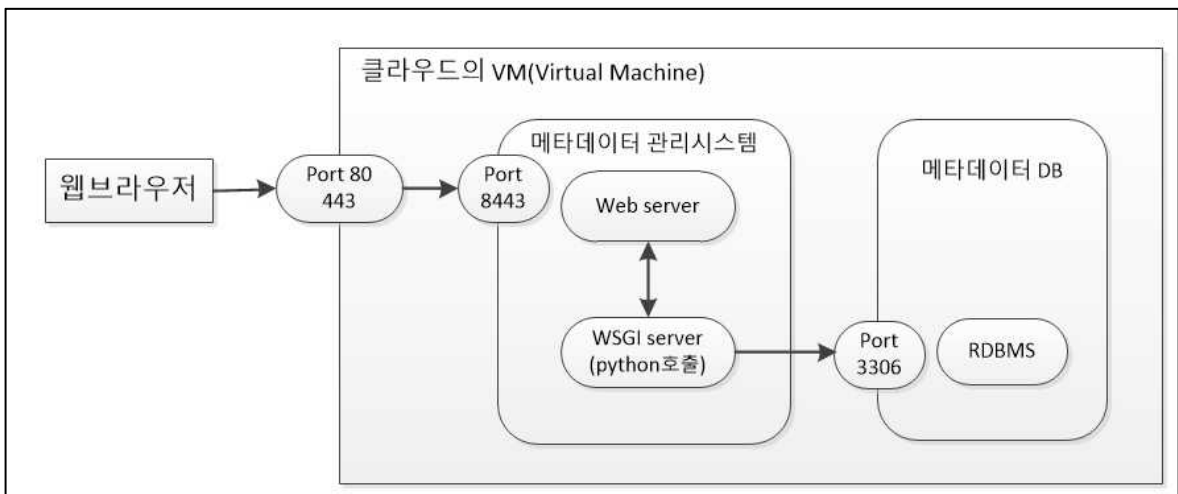
11.1. 목적

- 자율주행 통합교통관제센터 표준검증플랫폼은 통합관제시스템에 연동되는 시스템들의 표준 준용을 통해 전체 플랫폼의 안정성을 보장하고, 표준 적합성 테스트 환경을 구축, 범용적으로 표준적합성 테스트를 수행할 수 있는 기능을 제공한다.
- 본 운용자 매뉴얼은 표준검증플랫폼 서비스에 대한 운영 지침서를 작성하기 위하여 사용자, 그룹, 메타데이터 및 전용 컴파일러 관리를 웹 기능 중심으로 작성한다.
- 운용자 매뉴얼의 저작권은 개발사에 있으며, 개발사의 허락 없이 임의 수정을 할 수 없다.

11.2. S/W 운영 환경

11.2.1. 메타데이터 관리시스템의 운영 환경

- 메타데이터 관리시스템은 시스템의 운영 상 편리함과 확장성을 고려하여 Cloud(클라우드)의 VM(Virtual Machine) 환경에서 설치 및 운영되며 이를 간략하게 도식화하면 아래 그림과 같다.

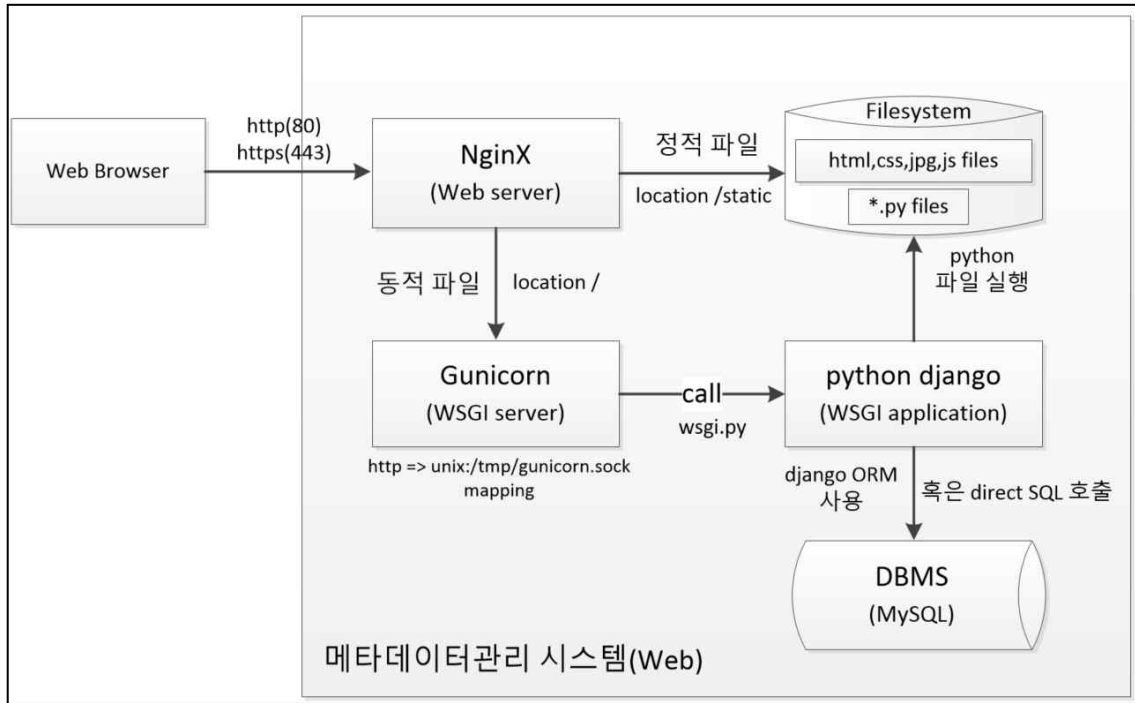


(그림 .101) 메타데이터 관리시스템의 운영 환경

- 그림과 같이 메타데이터 관리시스템의 내부에는 사용자의 웹브라우저 요청을 처리하는 Web server와 메타데이터의 business logic을 처리하는 python 코드, 마지막으로 양쪽을 상호 연동하는 WSGI(Web Server Gateway Interface) server가 존재한다.
- 또한 메타데이터 자체를 저장하기 위한 메타데이터 DB가 존재하며 이에 대한 S/W로 RDBMS(Relational Database Management System)가 사용된다.

11.2.2. 메타데이터 관리시스템의 논리적 S/W 블록 구조

- 아래 그림은 메타데이터 관리시스템의 논리적 S/W 블록 구조를 간략하게 도식화한 그림이다.

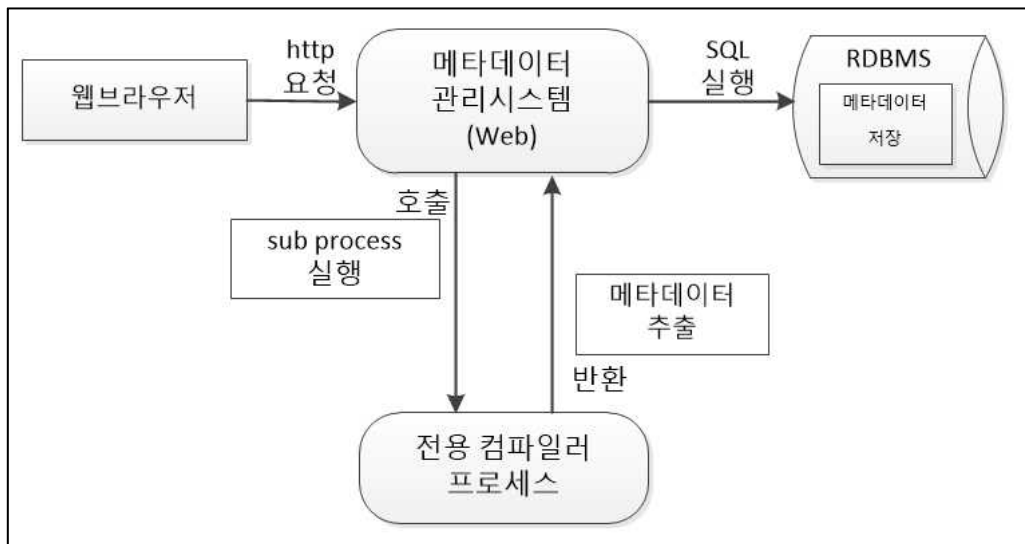


(그림 .102) 메타데이터 관리시스템 내부구조

- 사용자 웹브라우저로부터의 http(혹은 https) 요청을 처리하는 Web Server로서 NginX가 사용되고 있으며 이러한 사용자의 요청을 분석하여 해당 요청을 메타데이터 관련 모듈과 연결 및 처리하는 python web-framework(django)가 사용되며, Web Server와 python web-framework(django)를 상호 연결하는 웹 게이트웨이 서버로서 Gunicorn 서버가 사용되고 있다.
- 메타데이터의 저장, 추출, 검색 등에 사용되는 business logic에 대한 코드는 모두 python으로 작성되어 있으며, 메타데이터를 DBMS에 저장, 추출하기 위해서 python django의 ORM(Object-Relational Model) 인터페이스를 사용하여 실제 데이터를 저장 및 추출하게 된다.

11.2.3. 메타데이터 관리시스템과 전용 컴파일러 연동 구조

전용 컴파일러를 통해 메타데이터를 추출을 하기 위해 사용자가 메타데이터 관리시스템에 접속 및 추출 요청을 선택한 경우, 메타데이터 관리시스템은 해당 요청으로부터 사용자가 전달한 표준 문서(input)와 입력 인자(input parameter) 그리고 환경 설정 파일(configuration) 등을 command line 프로세스 호출을 위한 스트링으로 구성한 후 해당 스트링을 이용하여 sub process 방식으로 전용 컴파일러를 호출한다.



(그림.103) 메타데이터 관리시스템과 전용 컴파일러 연동 구조

이때 메타데이터 관리시스템은 전용 컴파일러에게 전달하는 입력 인자(입력 파일, 환경 설정 파일 포함)들을 command line parameter 전달 방식으로 전달하고, 이후 전용 컴파일러 프로세스는 메타데이터 추출을 완료한 경우 해당 메타데이터들을 STDOUT을 통해 출력 데이터들을 메타데이터 관리시스템에게 전달한다. 이렇게 추출된 메타데이터들은 메타데이터 관리시스템에서 SQL 구문 실행을 통해서 RDBMS의 메타데이터 DB에 저장된다.

11.3. S/W 설치

11.3.1. 메타필수 설치 조건(prerequisite)

○ 메타데이터 관리시스템은 특정 OS(운영체제)에 한정되어 개발 및 운영되도록 설계된 시스템이 아니지만 시스템의 운영 상 편리함과 확장성을 고려하여 Cloud(클라우드)의 VM(Virtual Machine) 환경에서 설치 및 운영되며 구체적으로 Ubuntu 16 server 환경에서 개발 및 운영이 최적화 되어 있다. 메타데이터 관리시스템이 사용하는 개발 환경 혹은 라이브러리 목록은 아래와 같다.

○ Python

메타데이터 관리시스템은 python으로 제작된 프로그램이므로 python 실행 환경 설치가 필수적이다.

○ Conda

Python을 위한 가상환경으로 시스템에 설치되어 있는 python 버전과는 별도로 다수의 여러 버전 python을 설치 및 구동할 수 있다. 개발 서버의 환경 구성과 상용 서버의 환경 구성을 동일하게 유지할 수 있으므로 개발과 디버깅이 쉽다.

○ Python Django

Python Django는 python으로 만들어진 MVC(Model-View-Controller) 기반의 웹 프레임워크(Web framework) 이며 django에서는 같은 개념을 MTV(Model - Template - View)라고 부른다. MVC 패턴은 데이터(model), 사용자 인터페이스(view), 데이터 처리 로직(controller)을 구분해 하나의 요소가 다른 요소에게 영향을 주지 않도록 설계하는 방식이다.

○ ORM

Object-relational mapping(객체 관계 매핑)의 약자로 데이터베이스의 테이블을 객체화하여, 각 DBMS에 대해서 CRUD(Create,Read,Update,Delete) 등을 공통된 접근기법으로 사용할 수 있다.

○ MySQL

1995년에 발표된 오픈 소스 RDBMS로서 관계형 데이터베이스를 관리하는 시스템으로 테이블 행(컬럼) 단위로 검색, 삽입, 갱신, 삭제를 쉽게 해주는 관리 소프트웨어이다. MySQL 5.7.8부터 DBMS 차원의 JSON 데이터를 지원하여 반정형(Semi-structured) 데이터를 테이블에 저장할 수 있게 되었다.

○ NginX(Web Server)

NginX는 웹서버 소프트웨어로, 가벼움과 높은 성능을 가지고 있다. 웹서버는 일반적으로 미리 만들어진 html 파일(정적파일)을 다수의 접속자에게 동시에 서비스할 수 있도록 설계되어 있다.

○ Gunicorn(WSGI Server)

특정 언어로 된 코드를 웹서버와 연동하기 위해서는 서로 통신할 수 있는 게이트웨이(gateway) 역할을 하는 매개체가 필요한데 그 다리 역할을 하는 것이 WSGI(Web

Server Gateway Interface) Server이다. Gunicorn은 웹서버와 python을 연동해 주는 소프트웨어이다.

11.3.2. Ubuntu 필수 패키지 설치

- Ubuntu 16 server에 필수적으로 설치되어야 할 패키지는 아래와 같으며 설치 시 root 권한이 필요하다.

```
sudo apt update
sudo apt-get update
sudo apt-get install wget python-dev python3-venv libmysqlclient-dev gcc -y
```

11.3.3. Conda 설치

```
# python 가상환경 conda 설치
mkdir -p ~/miniconda3
wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh -O
~/miniconda3/miniconda.sh
bash ~/miniconda3/miniconda.sh -b -u -p ~/miniconda3
~/miniconda3/bin/conda init bash
~/miniconda3/bin/conda init zsh

# python 3.8 가상환경 생성
source ~/.bashrc && ~/miniconda3/bin/conda create -y -n python38 python=3.8
cd ~ && source ~/miniconda3/etc/profile.d/conda.sh && conda activate python38
```

11.3.4. MySQL 설치

- 메타데이터 관리시스템이 사용하는 RDBMS는 MySQL이며 아래와 같이 설치할 수 있다. 구체적인 설치 및 설정은 이 문서의 설명 범위를 넘는 관계로 MySQL 공식 문서를 참고하여 설치 및 설정한다.(<https://dev.mysql.com/doc/>)

```
sudo apt-get install mysql-server # mysql server 설치
sudo service mysql start # mysql server 실행
```

11.3.5. NginX 설치

```
# NginX 설치. root 권한이 필요
sudo apt-get update
sudo apt-get install nginx

# firewall 설정
sudo ufw app list # Output 에 Nginx HTTP/HTTPS 확인
```

Output

Available applications:

Nginx Full
Nginx HTTP
Nginx HTTPS
OpenSSH

firewall port 추가 및 확인

```
sudo ufw allow 'Nginx Full'  
sudo ufw status
```

Output

Status: active

To	Action	From
--	-----	-----
Nginx Full	ALLOW	Anywhere
Nginx Full (v6)	ALLOW	Anywhere (v6)

11.3.6. 메타데이터 관리시스템 설치

- 메타데이터 관리시스템은 설치 스크립트(install.sh) 형태로 제공되며 이는 제공되는 USB 메모리에서 작업자의 PC로 복사하여 이를 클라우드의 VM 서버에 scp 명령어를 사용하여 upload 한 후 클라우드의 VM 서버에서 해당 스크립트를 실행하여 메타데이터 관리시스템을 설치할 수 있다

```
C:\WUsers\WGITSN>scp install.sh gitsn@175.106.97.18:/home/gitsn # gitsn 사용자의 home  
디렉토리로 업로드  
./install.sh # 설치 스크립트 실행. /home/gitsn/django/itsk에 메타데이터 관리시스템이 설치 됨
```

11.3.7. python 패키지 설치

- 메타데이터 관리시스템이 내부적으로 사용하는 python 패키지들을 설치하는 과정이며 메타데이터 관리시스템 설치 후 해당 디렉토리에 존재하는 requirements.txt 파일에 해당 패키지 목록이 존재하며 아래 명령어를 통해 한번에 설치할 수 있다

```
cd ~/django/itsk # 메타데이터 관리시스템 설치 경로
```

```
pip install -r requirements.txt # 메타데이터 관리시스템이 사용하는 python 패키지 설치
```

11.4. S/W 설정 및 구동

11.4.1. 메타데이터 관리시스템 환경 설정

- 메타데이터 관리시스템이 외부 시스템과 연동하여 동작할 때 해당 외부 시스템에 대한 정보를 설정할 필요가 있다. 해당 파일은 보안상의 이유로 패키지 배포와는 별도의 파일 (.env 파일)로 설치 및 설정해야 하며 주로 데이터를 저장, 추출에 사용되는 Database(RDBMS) 접속 및 인증 정보와 사용자 인증에 사용되는 Email 접속 및 인증 정보를 설정하게 된다. 설정이 필요한 항목과 각 항목의 내용은 아래 내용을 참조하여 설정한다.

```
cd django/itsk
vi .env # 메타데이터 관리시스템이 사용하는 외부 시스템 설정 정보 파일

# 아래 내용은 .env 파일의 내용
DB_NAME=[DB이름] # Database 이름
DB_USER=[DB사용자이름] # Database 사용자 계정
DB_HOST=localhost
DB_PASSWORD=[DB사용자암호] # Database 사용자 암호
EMAIL_HOST=[EMAIL호스트] # Email(SMTP) server 호스트 네임 혹은 ip
EMAIL_PORT=[EMAIL포트] # Email(SMTP) server 호스트 port
EMAIL_USER=[EMAIL사용자이름] # 사용자 Email 계정
EMAIL_PASSWORD=[EMAIL사용자암호] # 사용자 Email 암호
EMAIL_NAME=[EMAIL발신사용자주소] # 발신 사용자 email 주소
```

11.4.2. NginX 환경 설정 및 구동 스크립트

- 사용자가 메타데이터 관리시스템에 접속 및 특정 요청을 하는 경우 메타데이터 관리시스템은 내용이 변하지 않는 정적 html의 경우 NginX 웹서버가 사용자에게 직접 해당 파일을 전달하고 그 외 내용이 변하는 동적 스크립트를 사용자가 요청하는 경우 Unicorn Server가 해당 python 스크립트를 실행한 후 해당 결과를 사용자에게 전달하는 구조를 가지고 있다. 따라서 이러한 내용을 NginX 웹서버의 설정 파일에 반영해 주는 작업이 필요하다.

```
cd /etc/nginx/sites-enabled # NginX 환경 설정 디렉토리

sudo vi metamgt # 메타데이터 관리시스템 연결 설정 파일. root 권한이 필요

server_tokens off;
server {
    listen 80 default_server;
```

```

    return 444;
}
server {
    server_name .metadatamgt.com;
    listen 80;
    return 307 https://$host$request_uri;
}
server {
    listen 443 ssl;
    index index.html index.htm;
    client_max_body_size 500m;
    ssl_certificate /etc/nginx/sites-available/server.crt;
    ssl_certificate_key /etc/nginx/sites-available/server.key;
    ssl_protocols TLSv1 TLSv1.2;
    ssl_prefer_server_ciphers on;
    access_log /var/log/nginx/its_ssl_access.log;
    error_log /var/log/nginx/its_ssl_error.log;
    location = /favicon.ico { access_log off; log_not_found off; }
    location /static { # 메타데이터 관리시스템의 정적 파일들이 존재하는 경로
        alias /home/gitsn/django/its/static;
    }
    location /media { # 메타데이터 관리시스템의 media 파일들이 존재하는 경로
        alias /home/gitsn/django/its/media;
    }
    location / { # Gunicorn Server와의 socket 연결 설정
        include proxy_params;
        proxy_pass http://unix:/tmp/gunicorn.sock;
    }
}

```

```

# NginX 시작
sudo systemctl start nginx
# NginX 정지
sudo systemctl stop nginx
# NginX 재시작
sudo systemctl restart nginx
# NginX 환경설정 reloading
sudo systemctl reload nginx

```

11.4.3. Gunicorn 환경 설정 및 구동 스크립트

- 앞서 설명한 바와 같이 메타데이터 관리시스템은 python 으로 작성된 시스템이며 웹브라우저에서 http 요청이 들어올 경우 이를 python 으로 작성된 메타데이터 관리시스템과 연

동해 줄 gateway 역할이 필요한 데 이러한 역할을 하는 S/W가 Gunicorn (WSGI Server)이다. Gunicorn은 상시 동작해야 하므로 아래와 같이 Ubuntu system service에 설치 및 등록하여 사용하며 먼저 설치된 메타데이터 관리시스템과의 연동 위치 등을 설정해 줄 필요가 있다.

```
vi /etc/systemd/system/gunicorn.service
```

```
# 아래는 gunicorn 서비스 설치 및 설정 파일
```

```
[Unit]
```

```
Description=gunicorn daemon
```

```
After=network.target
```

```
[Service]
```

```
User=gitsn # gunicorn 동작 사용자 계정
```

```
Group=gitsn
```

```
WorkingDirectory=/home/gitsn/django/its # 메타데이터 관리시스템 경로
```

```
EnvironmentFile=/home/gitsn/django/its/its.env
```

```
ExecStart=/home/gitsn/miniconda3/envs/python38/bin/gunicorn W
```

```
    --workers 4 W
```

```
    --reload W
```

```
    --access-logfile /var/log/gunicorn/access.log W
```

```
    --error-logfile /var/log/gunicorn/error.log W
```

```
    --bind unix:/tmp/gunicorn.sock W # NginX 연결 socket 설정
```

```
    its.wsgi:application
```

```
[Install]
```

```
WantedBy=multi-user.target
```

```
# gunicorn.service 파일을 수정한 후 반드시 실행
```

```
systemctl daemon-reload
```

```
# gunicorn 서비스 시작
```

```
systemctl start gunicorn
```

```
# gunicorn 서비스 정지
```

```
systemctl stop gunicorn
```

```
# gunicorn 서비스 재시작
```

```
systemctl restart gunicorn
```

11.4.4. 메타데이터 관리시스템의 관리자 계정 생성

- 메타데이터 관리시스템의 설치 및 설정이 완료되고 NginX, Gunicorn 서버가 설치 및 설정이 완료되면 메타데이터 관리시스템을 웹 인터페이스를 통해 사용할 수 있다.
- 메타데이터 관리시스템이 처음 설치된 이후 시스템의 관리자 계정을 추가해 줄 수 있는데 아래와 같이 command-line 방식을 통해 해당 계정을 추가해 줄 수 있다. 해당 계정은 이후 웹 인터페이스를 통해 해당 계정에 대한 암호와 이메일 등 기타 정보를 변경해 줄 수 있다.

```
conda activate python38 # python v3.8 버전의 가상 환경 실행
cd ~/django/itsk # 메타데이터 관리시스템 설치 경로로 이동

# 메타데이터 관리시스템의 관리자 계정 등록 요청
python manage.py createsuperuser --settings its.settings-prod

사용자 이름 (leave blank to use 'gitns'): [계정] # 관리자 계정 입력
이메일 주소: [이메일] # 관리자 이메일 입력
Password: [암호] # 관리자 암호 입력
Password (again): [암호] # 관리자 암호 재입력
Superuser created successfully.
```

11.4.5. 메타데이터 관리시스템의 구동 스크립트

- 메타데이터 관리시스템은 Gunicorn 서버를 통해 구동되기 때문에 Gunicorn 서비스의 시작, 정지 스크립트를 통해 메타데이터 관리시스템을 시작, 정지시킬 수 있다. 즉 아래와 같이 gunicorn 서비스 시작/정지/재시작 요청을 통해 메타데이터 관리시스템의 시작/정지/재시작을 할 수 있다.

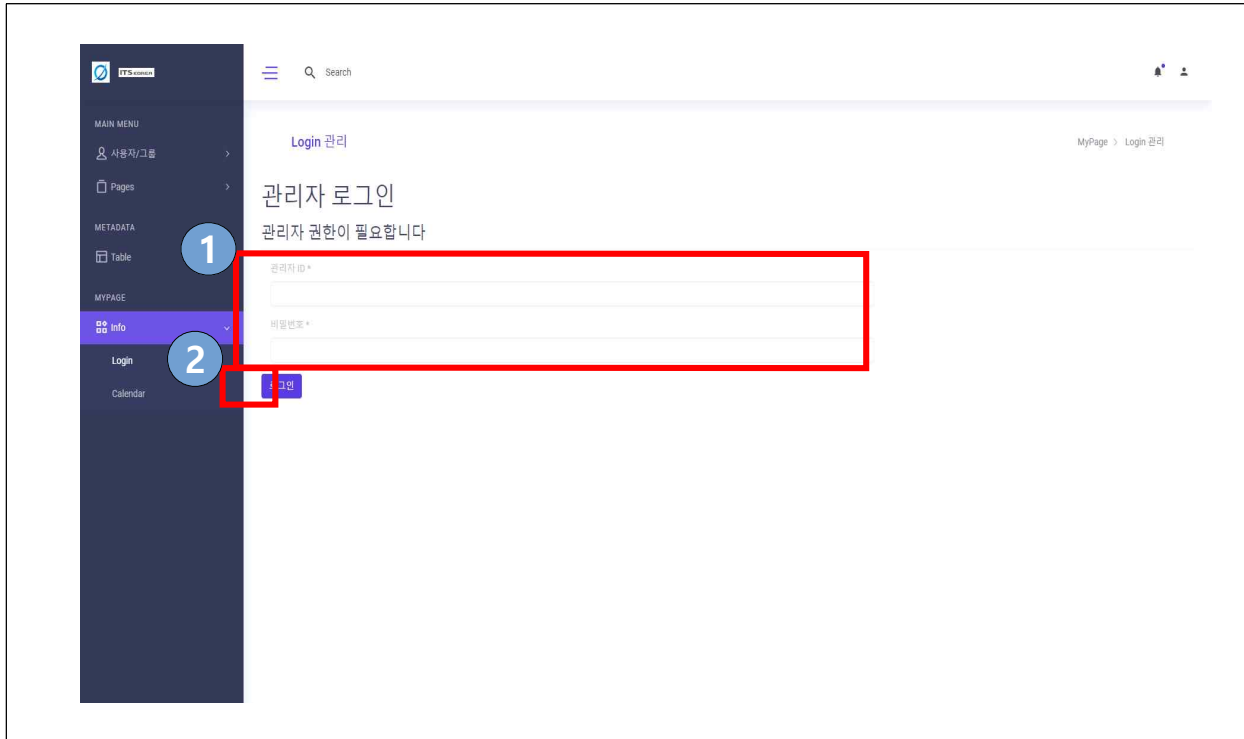
```
# gunicorn 서비스 시작
systemctl start gunicorn

# gunicorn 서비스 정지
systemctl stop gunicorn

# gunicorn 서비스 재시작
systemctl restart gunicorn
```

11.5. 운영자 화면 매뉴얼

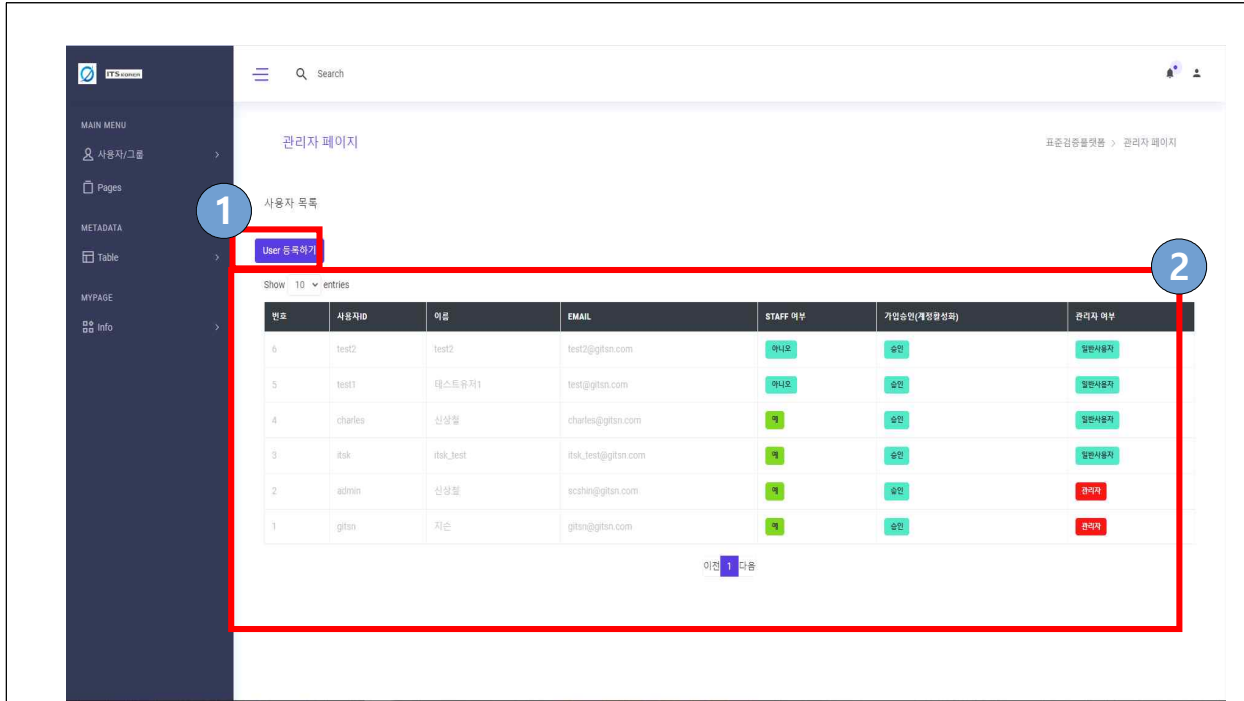
11.5.1. 로그인



화면 개요	웹 페이지 접속 후 첫 화면 [로그인]	
화면 설명	1	관리자 권한이 부여된 ID만 로그인 가능
	2	관리자 ID, 비밀번호 기재 후 '로그인' 버튼 클릭
사용 방법	1. 웹 주소 > 로그인 페이지 2. 초기화면 - 로그인 양식 화면 호출 3. 선택화면 - 관리자 아이디일 시 로그인 가능	
특기 사항	일반 사용자 아이디로는 로그인 불가능	

(표.28) 로그인

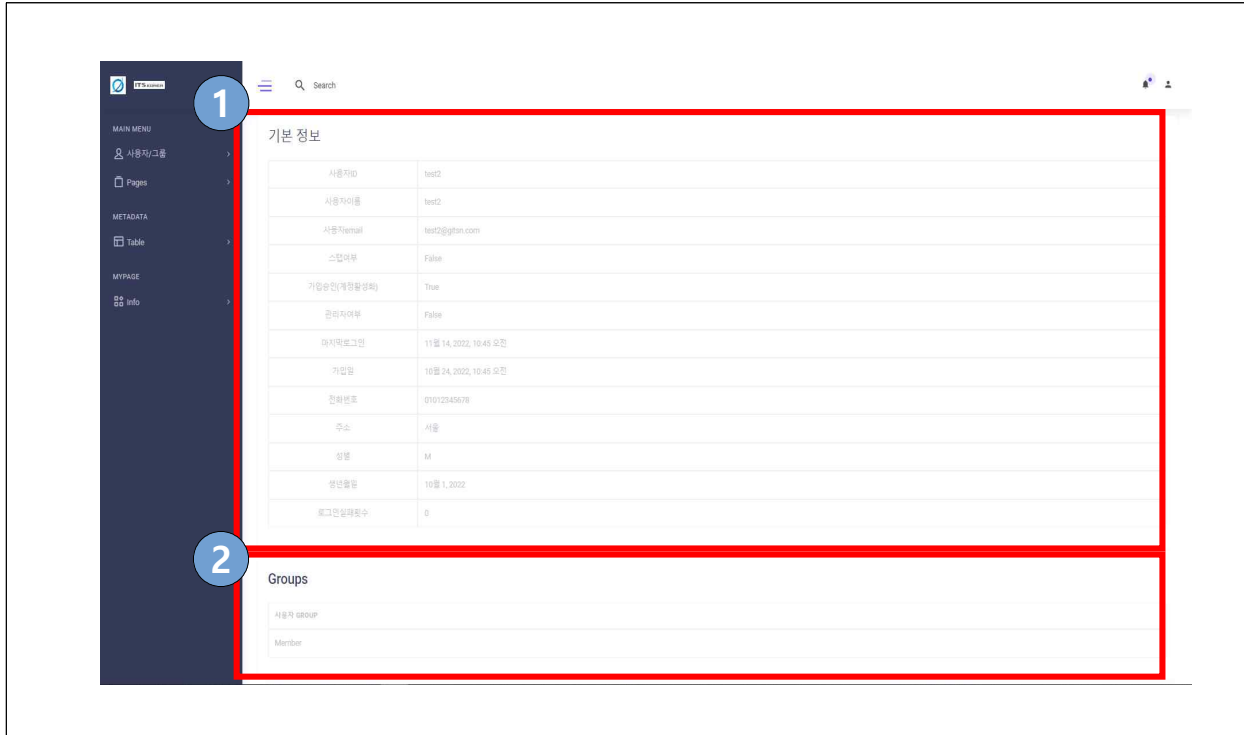
11.5.2. 사용자 관리



화면 개요	사용자 관리 화면	
화면 설명	1	사용자를 등록할 수 있으며 사용자가 회원가입 요청하는 것보다 빠르게 생성 가능하며 권한 부여도 할 수 있음
	2	전체 사용자 목록 조회 [회원가입 요청한 사용자 포함]
사용 방법	1. 메뉴 : 사용자/그룹 > 사용자 ID 선택 시 2. 초기화면 - 관리자 페이지의 메인 페이지 - 사용자 목록 조회 3. 선택화면 - 선택 시 사용자의 정보 조회 가능	
특기 사항	일반 사용자가 회원가입을 하는 경우 최초 가입승인 상태가 미승인 상태임	

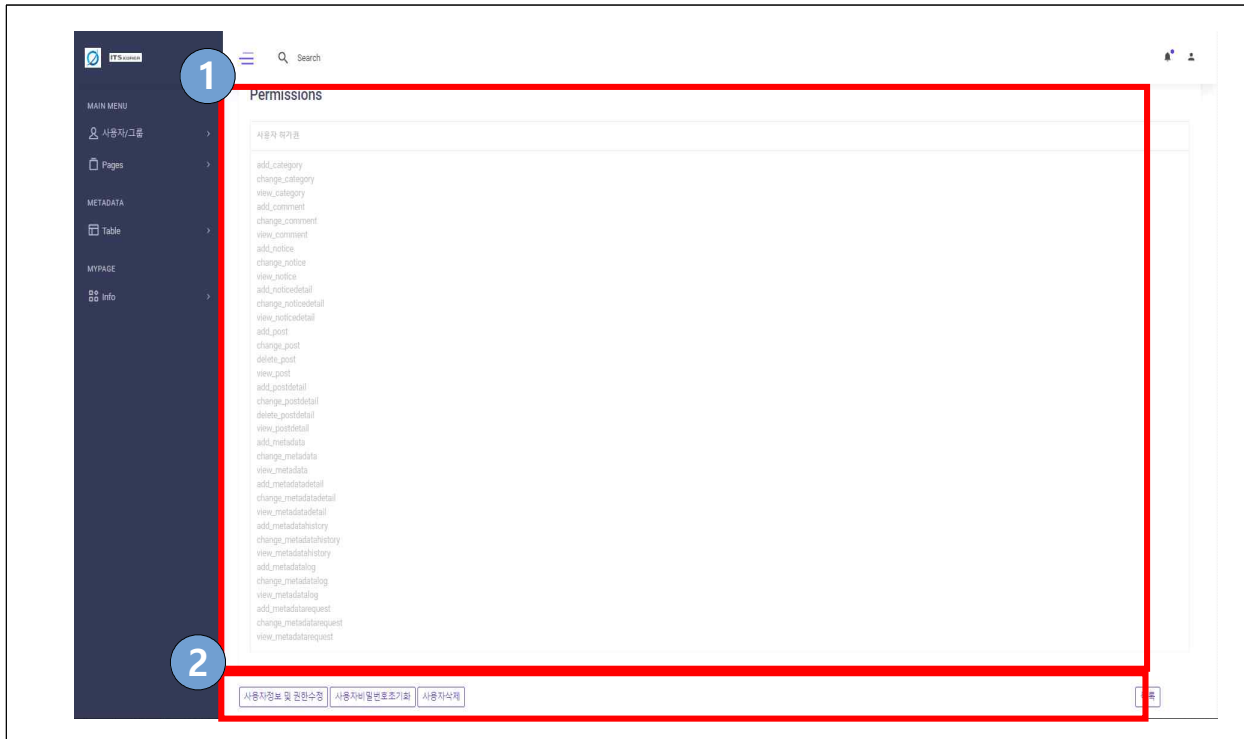
(표.29) 사용자 관리

11.5.3. 사용자 정보 조회



화면 개요	사용자 정보 조회 화면 #1	
화면 설명	1	등록된 기본 정보
	2	사용자가 포함되어있는 그룹
사용 방법	1. 메뉴 : 사용자 관리 > 사용자 정보 2. 초기화면 - 기본 정보 및 그룹, 허가권 확인 가능	
특기 사항	Groups 에는 사용자가 속한 그룹 목록이 표시	

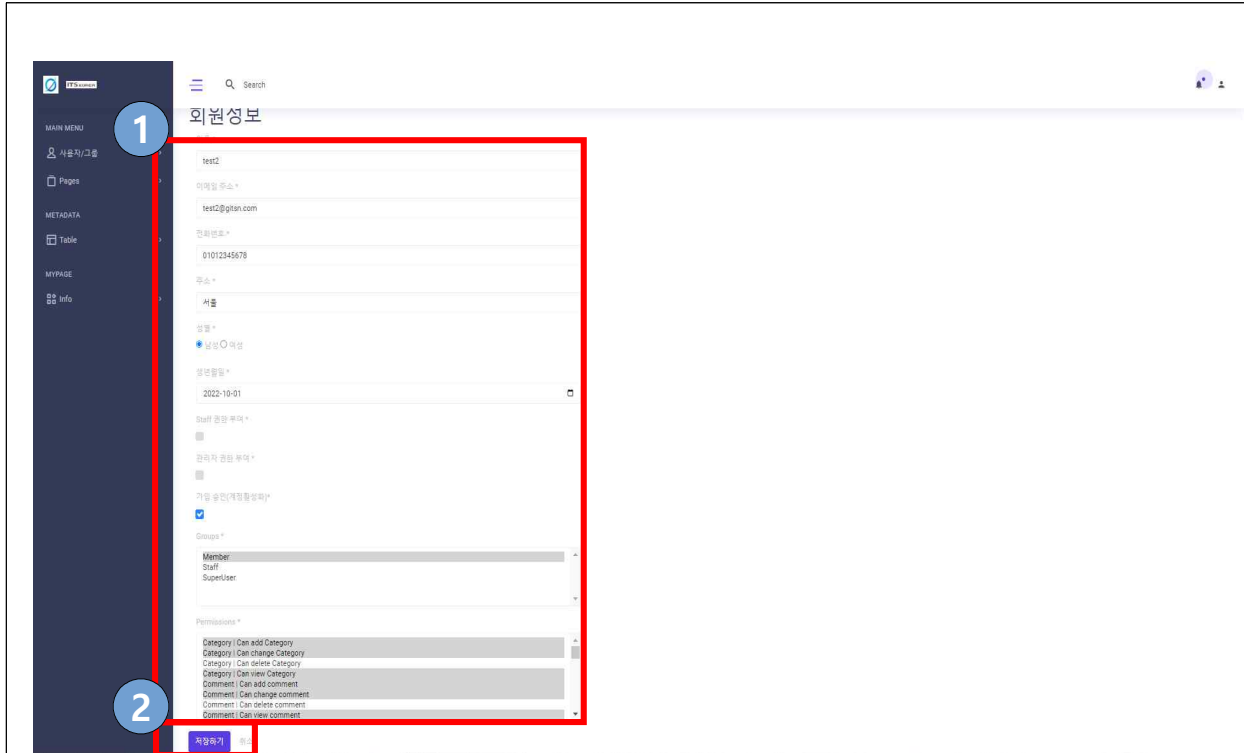
(표.30) 사용자 정보 조회 #1



화면 개요	사용자 정보 조회 화면 #2	
화면 설명	1	등록된 사용자의 허가권 정보
	2	사용자 정보 및 권한 수정, 비밀번호 초기화, 사용자 삭제 버튼 선택을 통해 해당 페이지로 이동 가능
사용 방법	1. 메뉴 : 사용자 관리 > 사용자 정보 2. 초기화면 - 기본 정보 및 그룹, 허가권 확인 가능, 정보 및 권한 수정, 비밀번호 초기화, 사용자 삭제 버튼을 통해 조작 가능 3. 선택화면 - 버튼을 통해 원하는 항목을 수정할 수 있음	
특기 사항	Permissions 에는 사용자에게 부여된 객체(Database 테이블)에 대한 권한의 종류가 나열됨 add_* : 테이블에 추가할 수 있는 권한 change_* : 테이블에 데이터를 수정할 수 있는 권한 delete_* : 테이블에 데이터를 삭제할 수 있는 권한 view_* : 테이블의 데이터를 조회할 수 있는 권한	

(표.31) 사용자 정보 조회 #2

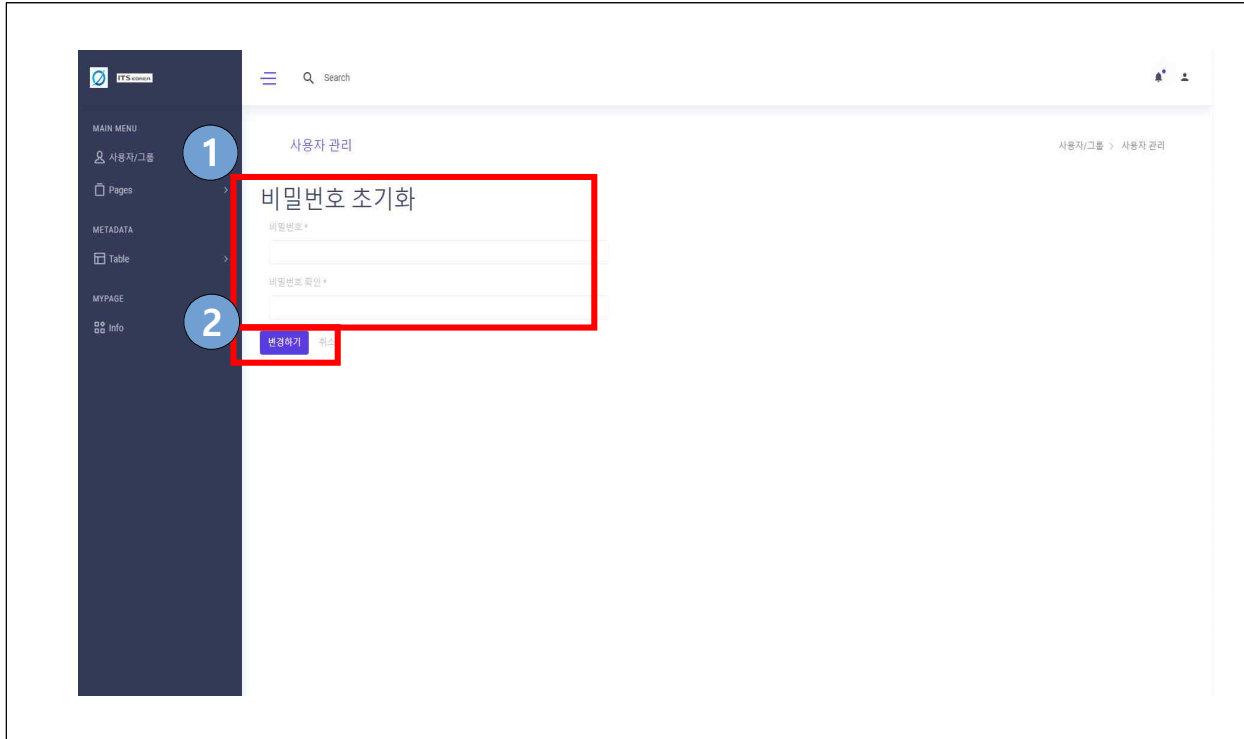
11.5.4. 사용자 정보 수정



화면 개요	사용자 정보 수정 화면	
화면 설명	1	등록된 사용자의 회원 정보
	1	staff, 관리자 권한 부여 체크 및 가입 승인 체크로 승인 및 부여 가능
	1	그룹 설정 및 허가권 설정 가능
	2	사용자 정보 수정 후 '저장하기' 버튼 클릭으로 변경
사용 방법	1. 메뉴 : 사용자 정보 > '사용자 정보 및 권한 수정' 버튼 2. 초기화면 - 사용자가 등록했던 회원 정보 표시 3. 선택화면 - 수정 후 버튼 클릭으로 변경	
특기 사항	Groups와 Permissions 항목을 선택할 경우 키보드의 'Ctrl' 키를 누른 상태에서 항목을 선택하면 복수 선택할 수 있음	

(표.32) 사용자 정보 수정

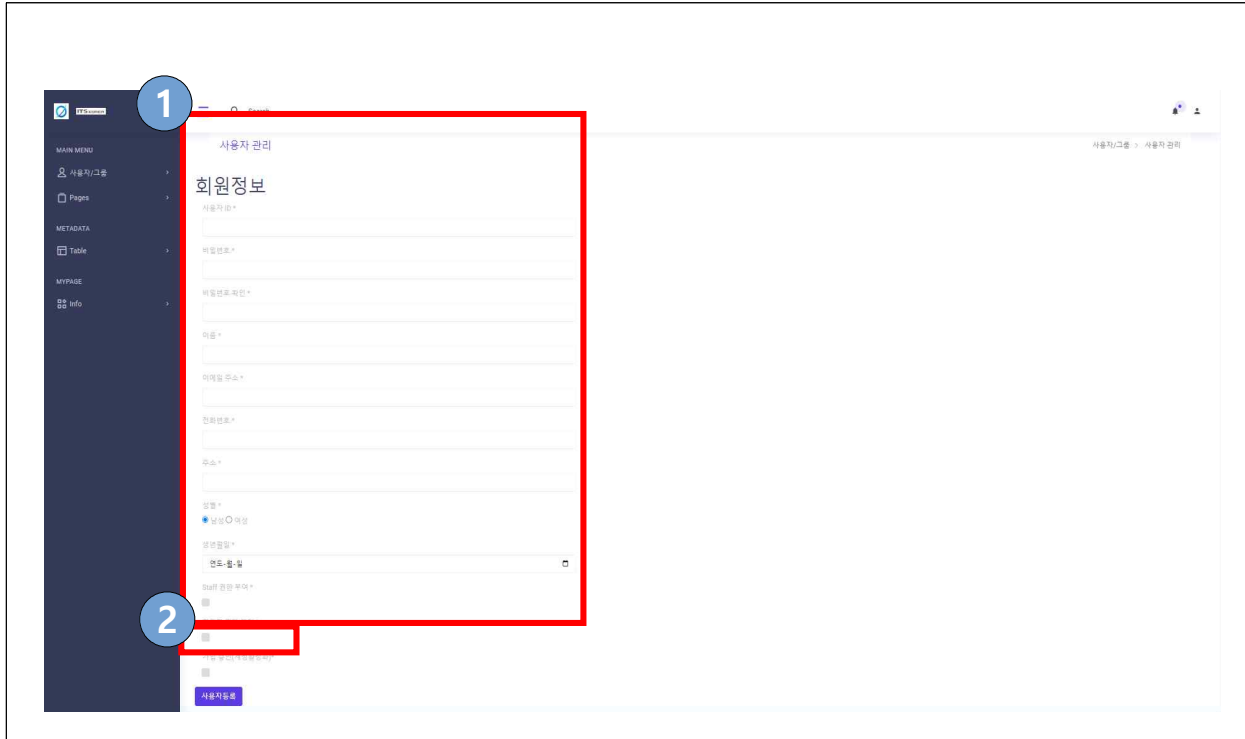
11.5.5. 비밀번호 초기화



화면 개요	비밀번호 초기화 화면	
화면 설명	1	새로 변경할 비밀번호를 두 번 입력
	2	‘변경하기’ 버튼
사용 방법	1. 메뉴 : 사용자 정보 > 비밀번호 초기화 2. 초기화면 - 새 비밀번호, 새 비밀번호 확인 입력 - ‘변경하기’ 버튼 클릭 후 변경	
특기 사항	두 번 입력한 비밀번호가 동일하지 않으면 ‘비밀번호가 일치하지 않습니다.’ 에러 표시됨	

(표.33) 비밀번호 초기화

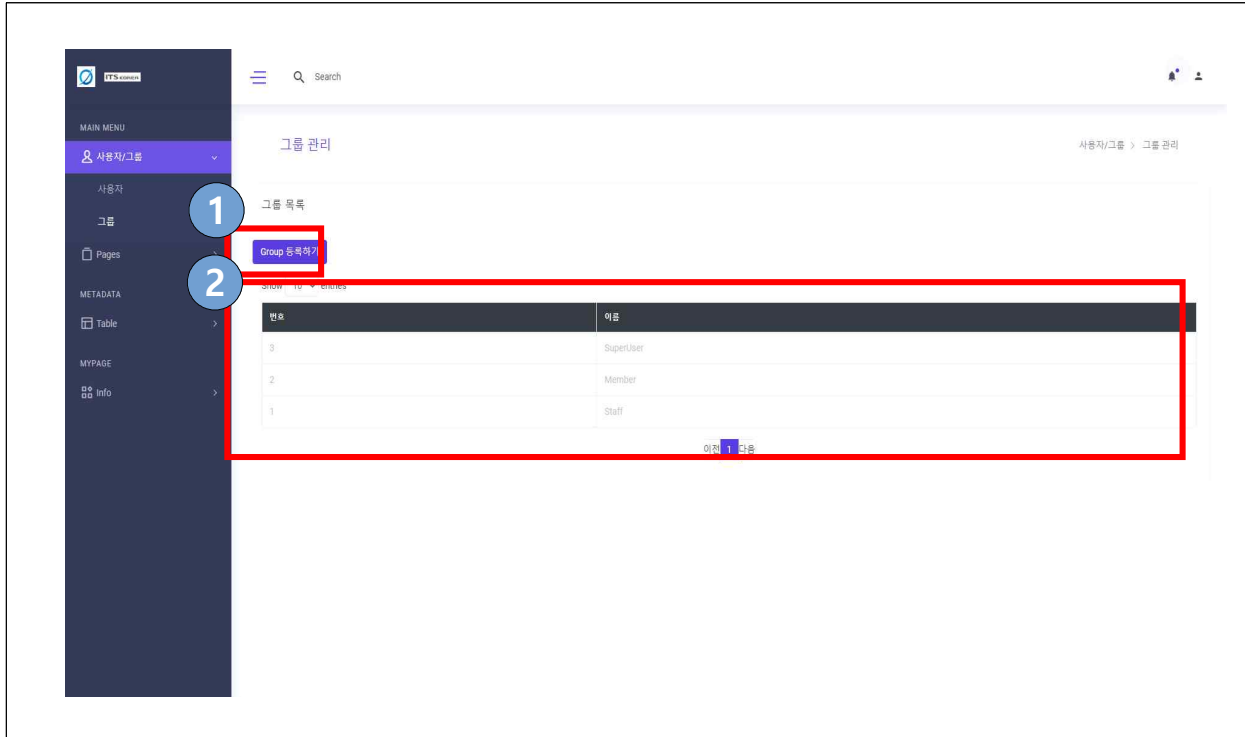
11.5.6. 사용자 등록



화면 개요	사용자 등록 화면	
화면 설명	1	회원 기본 정보 입력 창
	1	권한 부여 및 가입 승인으로 계정 활성화
	2	'사용자 등록' 버튼
사용 방법	1. 메뉴 : 사용자 > 'User 등록하기' 버튼 2. 초기화면 - 기본 정보 및 권한 부여, 가입 승인(계정활성화) 선택 3. 선택화면 - 버튼 클릭 후 생성	
특기 사항	가입 승인을 선택한 경우 회원가입 후 별도의 가입 승인을 할 필요 없음 동일한 사용자ID 혹은 이메일 주소가 존재하는 경우 '해당 사용자 이름이 이미 존재합니다.' 와 같은 에러가 화면에 표시되며 회원가입이 진행되지 않음	

(표.34) 사용자 등록

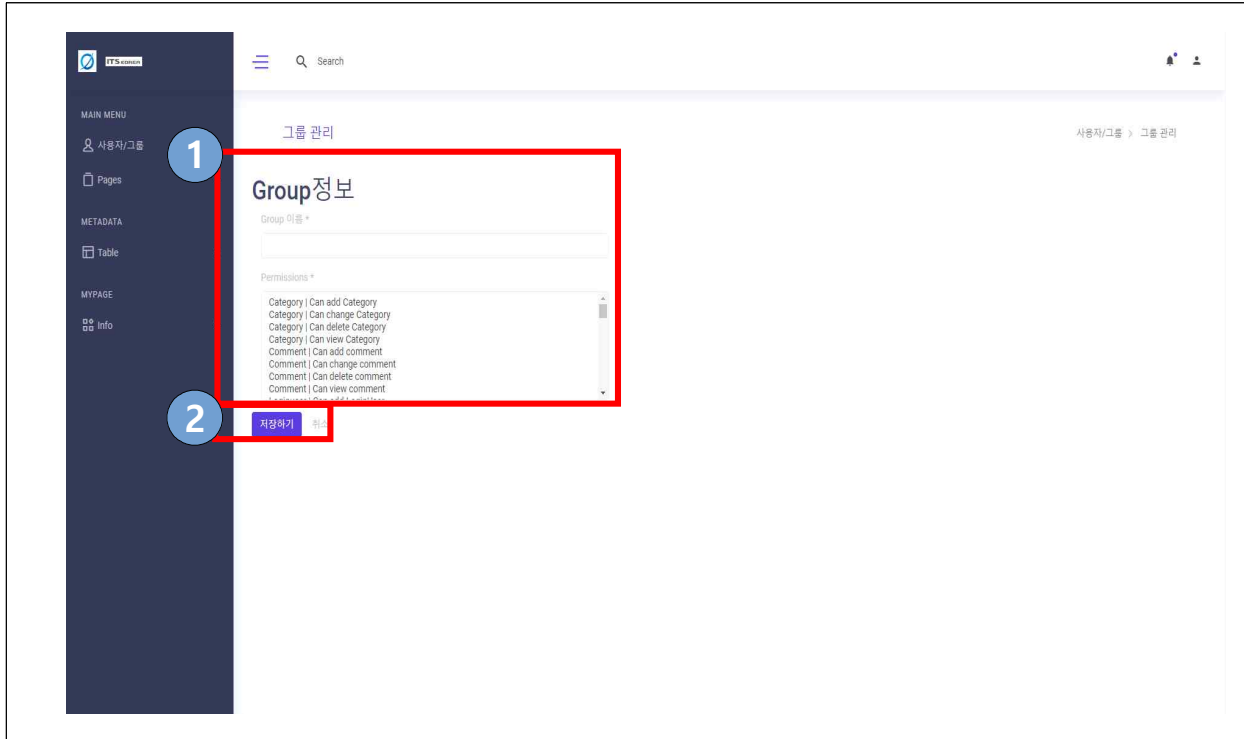
11.5.7. 그룹 관리



화면 개요	그룹 관리 화면	
화면 설명	1	그룹 등록 버튼
	2	전체 그룹 목록 조회
사용 방법	1. 메뉴 : 사용자/그룹 > 그룹 2. 초기화면 - 그룹 목록 조회 3. 선택화면 - 그룹 이름 선택 시 그룹의 정보 조회 가능	
특기 사항	없음	

(표.35) 그룹 관리

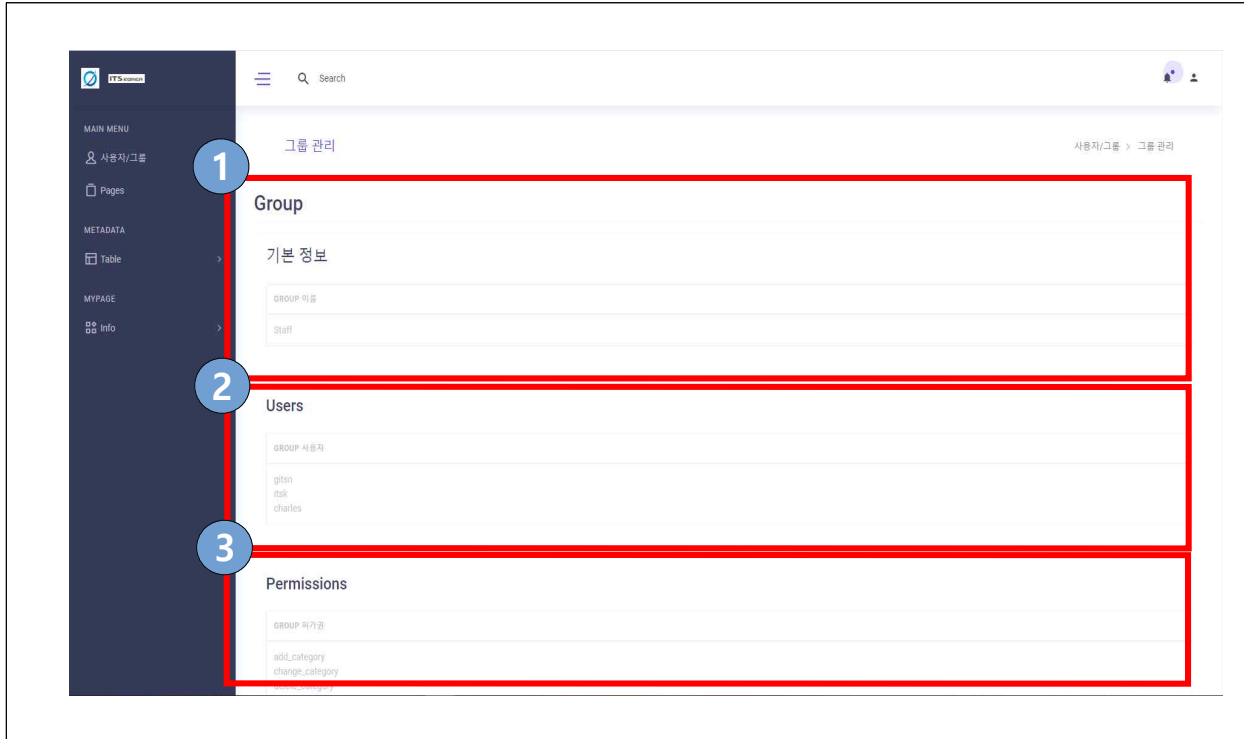
11.5.8. 그룹 등록



화면 개요	그룹 등록 화면
화면 설명	1 그룹 이름 입력 후 허가권 선택 부여 가능
	2 '저장하기' 버튼
사용 방법	<ol style="list-style-type: none"> 1. 메뉴 : 그룹 > 'Group 등록하기' 버튼 2. 초기화면 <ul style="list-style-type: none"> - 그룹 이름 부여 후 원하는 허가권을 선택해 그룹 생성 가능 3. 선택화면 <ul style="list-style-type: none"> - 저장하기 버튼 클릭 후 Group 생성
특기 사항	<p>Permissions 에는 Group에 부여된 객체(Database 테이블)에 대한 권한의 종류가 나열됨</p> <p>add_* : 테이블에 추가할 수 있는 권한</p> <p>change_* : 테이블에 데이터를 수정할 수 있는 권한</p> <p>delete_* : 테이블에 데이터를 삭제할 수 있는 권한</p> <p>view_* : 테이블의 데이터를 조회할 수 있는 권한</p> <p>Permissions 항목을 선택할 경우 키보드의 'Ctrl' 키를 누른 상태에서 항목을 선택하면 복수 선택할 수 있음</p>

(표.36) 그룹 등록

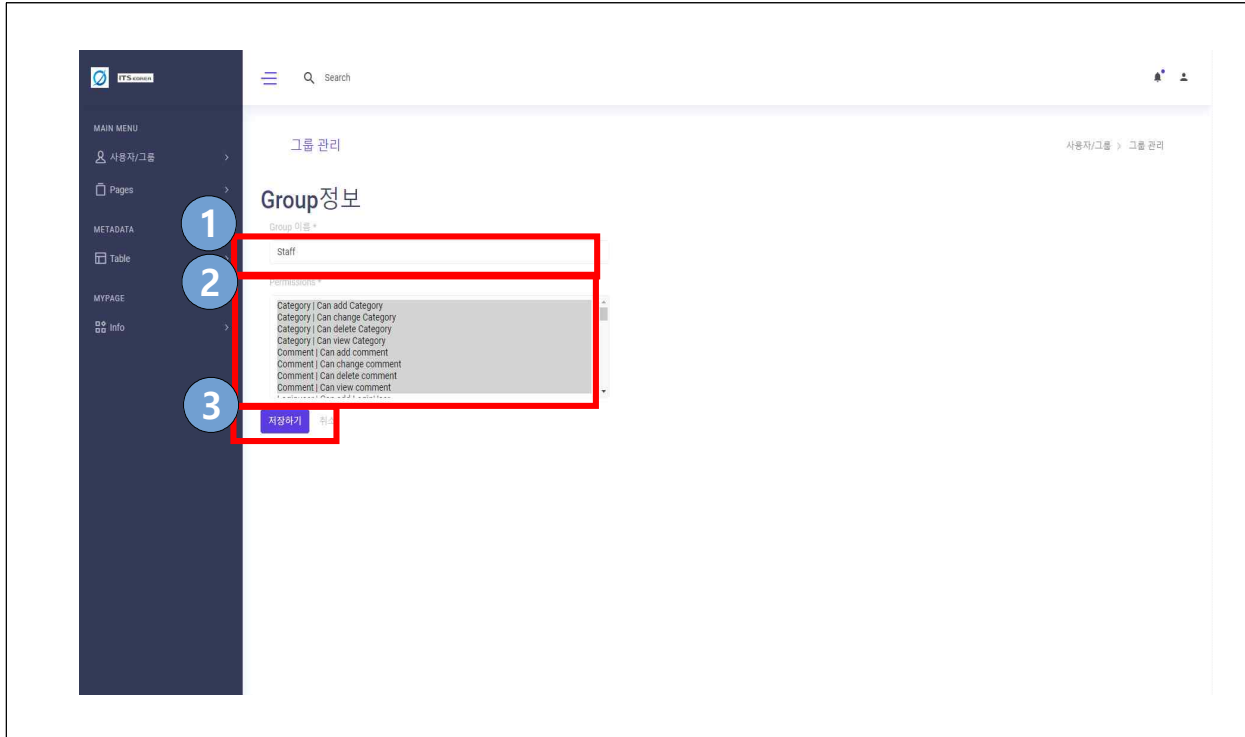
11.5.9. 그룹 정보 조회



화면 개요	그룹 정보 조회 화면	
화면 설명	1	그룹의 기본 정보
	2	그룹에 포함된 사용자 목록
	3	그룹이 가진 허가권 목록
사용 방법	1. 메뉴 : 그룹 > 그룹 정보 (목록에 있는 그룹)	
	2. 초기화면 - 그룹의 기본 정보, 그룹의 사용자, 그룹의 허가권 확인 가능 - 정보 하단에 정보를 수정할 수 있고 삭제할 수 있는 버튼이 있음	
특기 사항	3. 선택화면 - 수정, 삭제를 클릭하여 내용 수정 및 삭제 가능	
	Users에는 해당 그룹에 속한 사용자 목록이 표시 Permissions에는 Group에 부여된 객체(Database 테이블)에 대한 권한의 종류가 나열됨	

(표.37) 그룹 정보 조회

11.5.10. 그룹 정보 수정



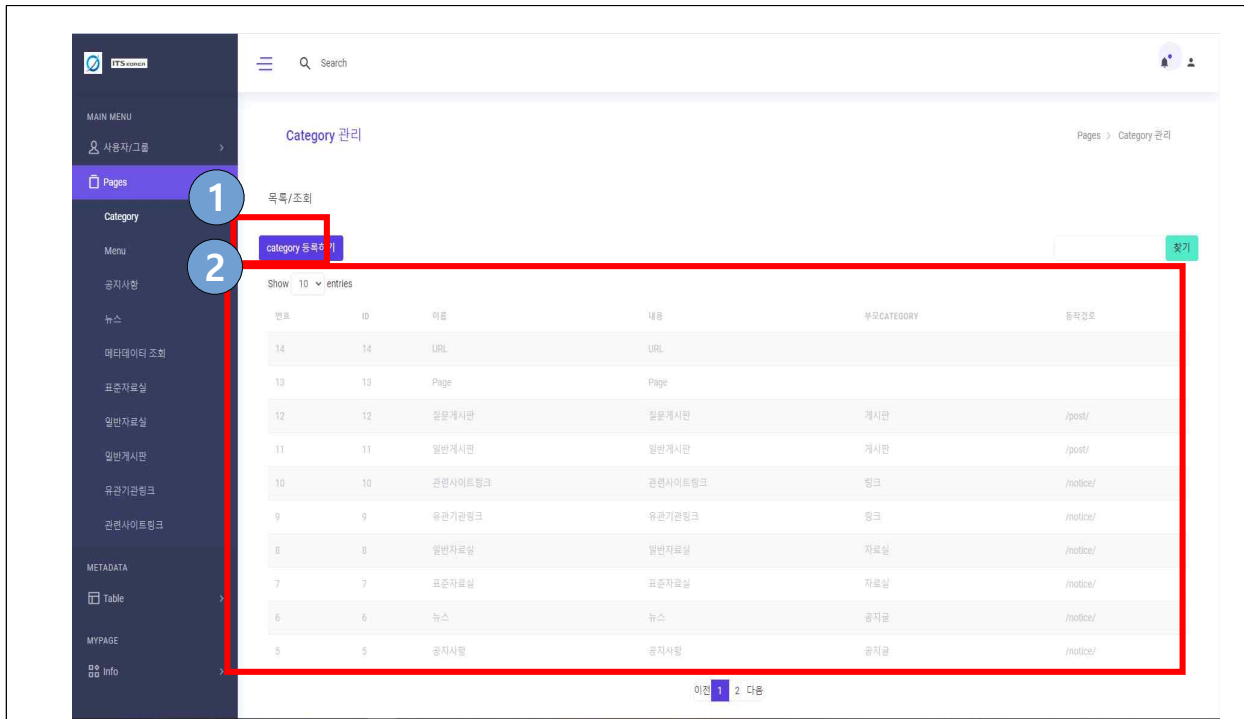
화면 개요	그룹 정보 수정 화면	
화면 설명	1	등록되어있는 그룹 이름
	2	선택되어있는 허가권
	3	'저장하기' 버튼
사용 방법	1. 메뉴 : 그룹 정보 > '수정' 버튼	
	2. 초기화면 - 그룹의 이름 및 허가 권한 수정 가능	
특기 사항	3. 선택화면 - 버튼 클릭 후 저장	
	Permissions 항목을 선택할 경우 키보드의 'Ctrl' 키를 누른 상태에서 항목을 선택하면 복수 선택할 수 있음	

(표.38) 그룹 정보 수정

-
-
-
-
-

-
-
-
-

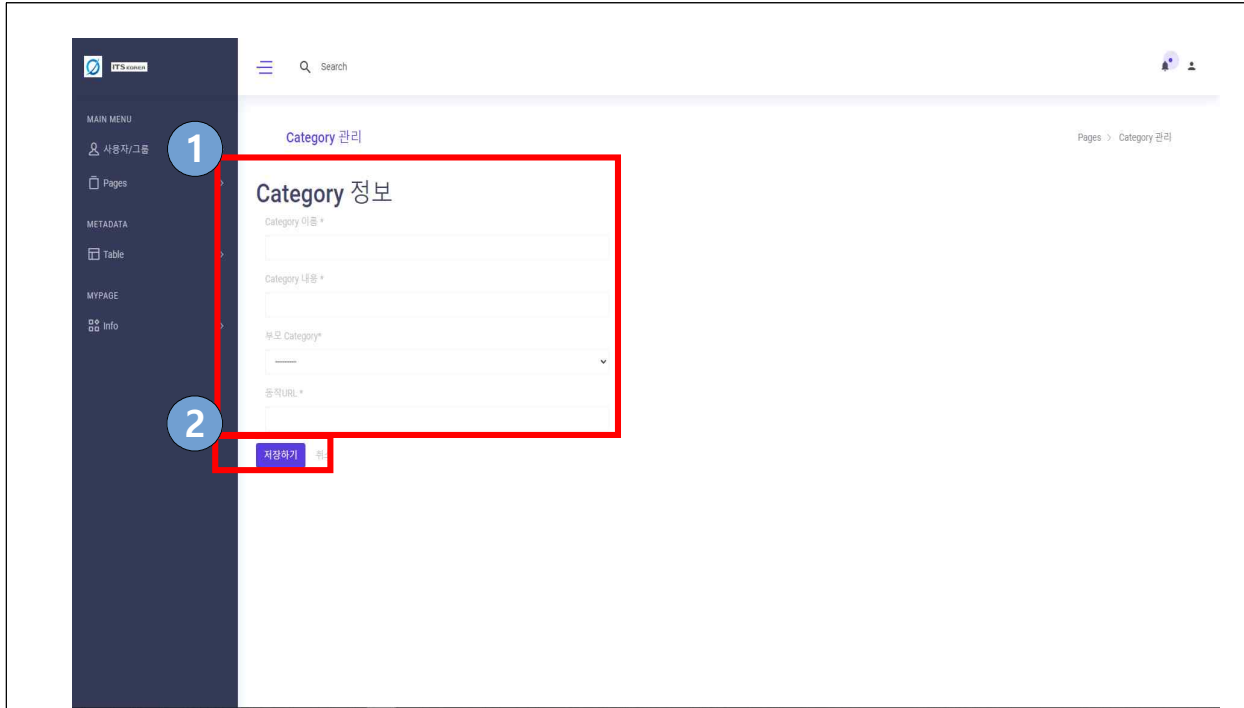
5.11 카테고리 관리



화면 개요	카테고리 관리 화면	
화면 설명	1	카테고리 등록 버튼
	2	전체 카테고리 목록 조회
사용 방법	1. 메뉴 : Pages > Category 2. 초기화면 - 카테고리 목록 조회 3. 선택화면 - Category 이름 선택 시 해당 카테고리의 정보 조회 가능	
특기 사항	Category는 게시판, 뉴스, 메뉴 등 특정 글의 분류를 나타냄	

(표.39) 카테고리 관리

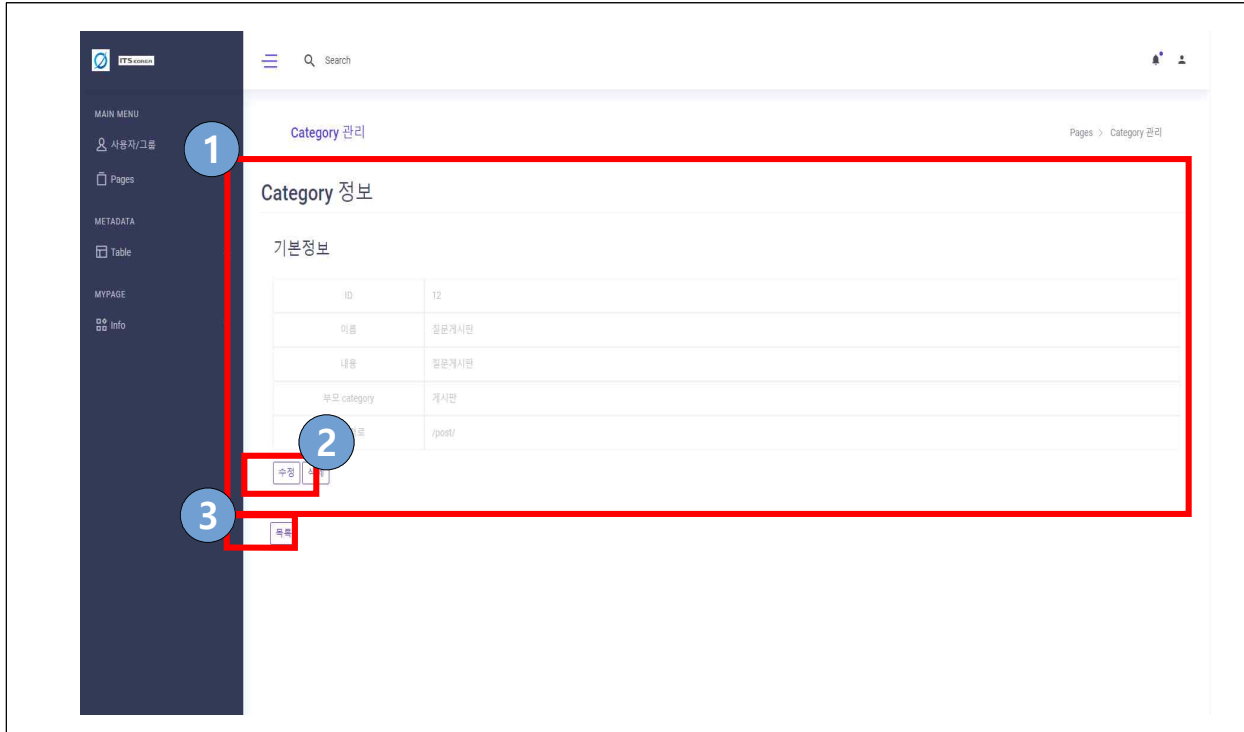
11.5.11. 카테고리 등록



화면 개요	카테고리 등록 화면	
화면 설명	1	카테고리 이름, 내용 입력란
	1	부모 카테고리 설정
	1	동작 URL 설정
	2	'저장하기' 버튼
사용 방법	1. 메뉴 : 카테고리 > 'Category 등록하기' 버튼	
	2. 초기화면 - 카테고리의 이름 및 내용, 부모 카테고리 설정 - 동작 URL 설정 (ex. /notice/)	
특기 사항	3. 선택화면 - 버튼 클릭 후 생성	
	동작 URL은 관리자만 글을 쓸 수 있는 공지글 성격의 경우 '/notice/' 를 입력하고 가입된 회원이 글을 읽고 쓸 수 있는 게시판 성격의 경우 '/post/' 를 입력 기타의 경우에는 link 될 url을 입력	

(표.40) 카테고리 등록

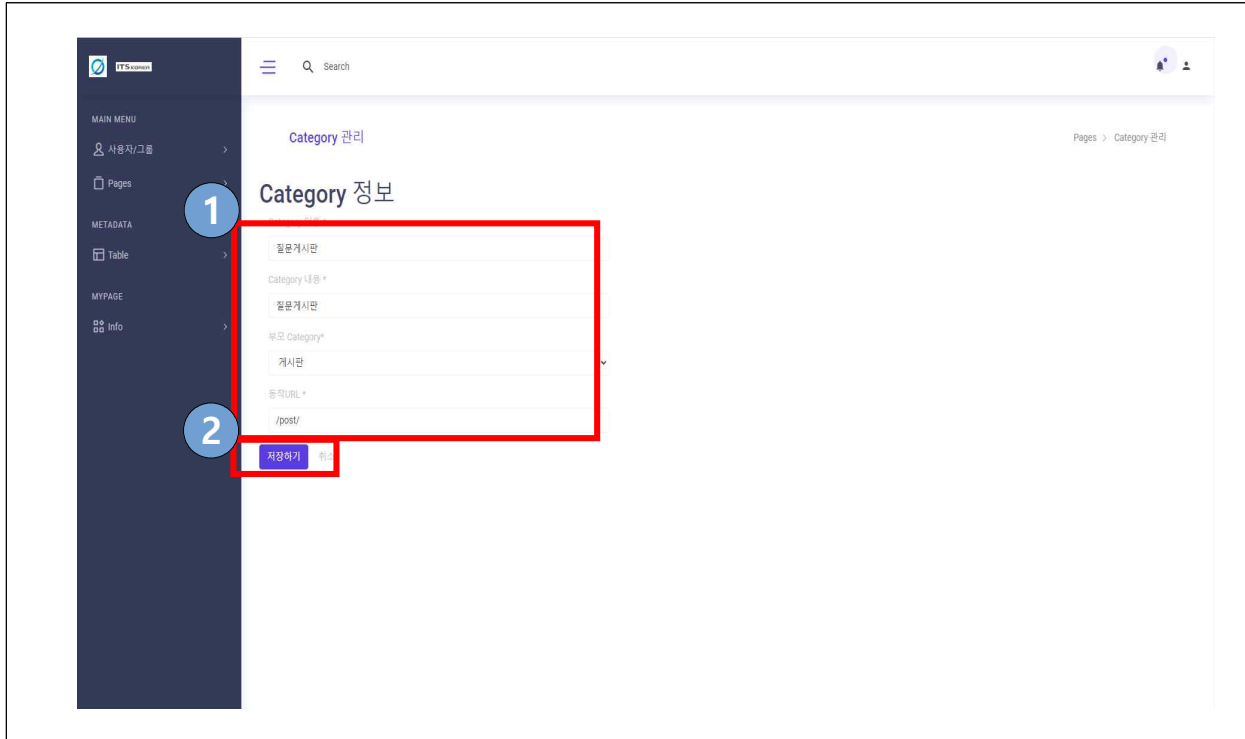
11.5.12. 카테고리 정보 조회



화면 개요	카테고리 정보 조회 화면	
화면 설명	1	카테고리의 기본 정보
	2	수정, 삭제 버튼
	3	'목록' 버튼
사용 방법	1. 메뉴 : 카테고리 목록 조회 > 카테고리 이름 선택 시 2. 초기화면 - 카테고리의 기본 정보 확인 가능 - 수정, 삭제 버튼을 통해 수정하고 삭제 가능 - 목록 버튼 클릭 시 카테고리 목록 조회 페이지로 이동	
특기 사항	없음	

(표.41) 카테고리 정보 조회

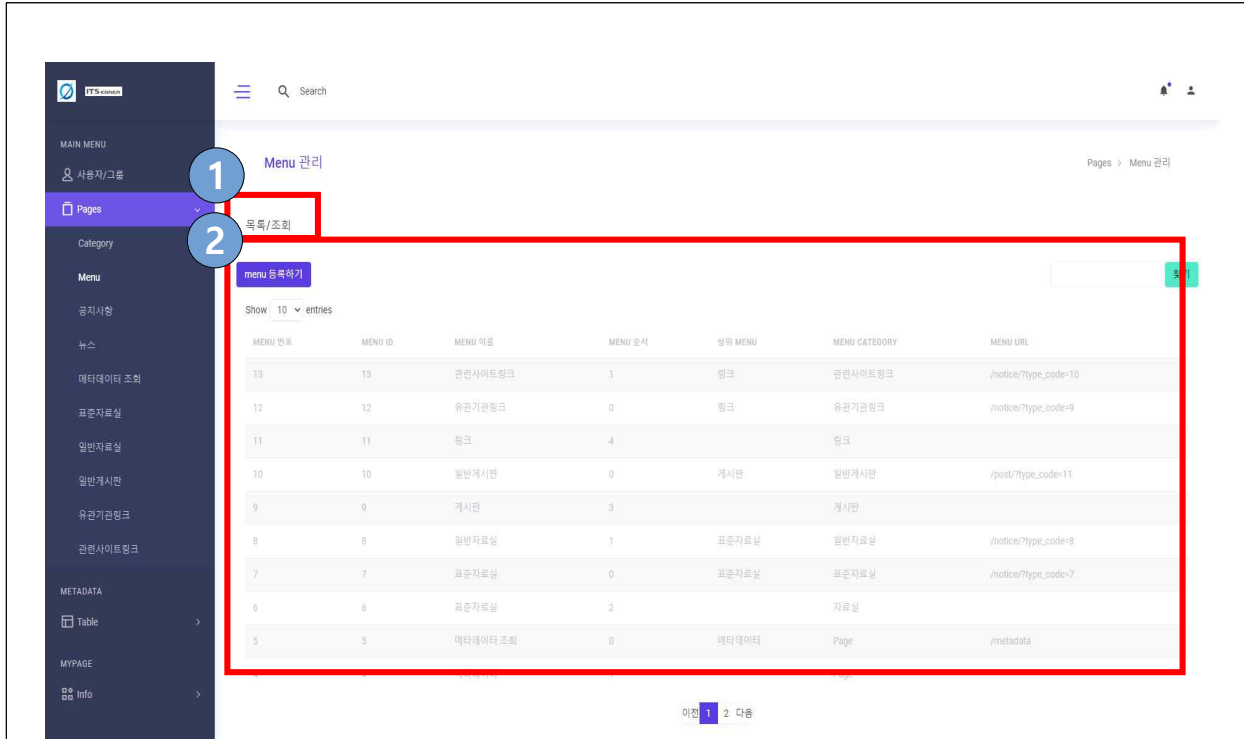
11.5.13. 카테고리 정보 수정



화면 개요	카테고리 정보 수정 화면	
화면 설명	1	등록되어있는 카테고리 기본 정보
	2	'저장하기' 버튼
사용 방법	1. 메뉴 : 카테고리 정보 > '수정' 버튼 2. 초기화면 - 카테고리의 기본 정보 수정 가능 3. 선택화면 - 저장하기 버튼 클릭 후 내용 저장	
특기 사항	없음	

(표.42) 카테고리 정보 수정

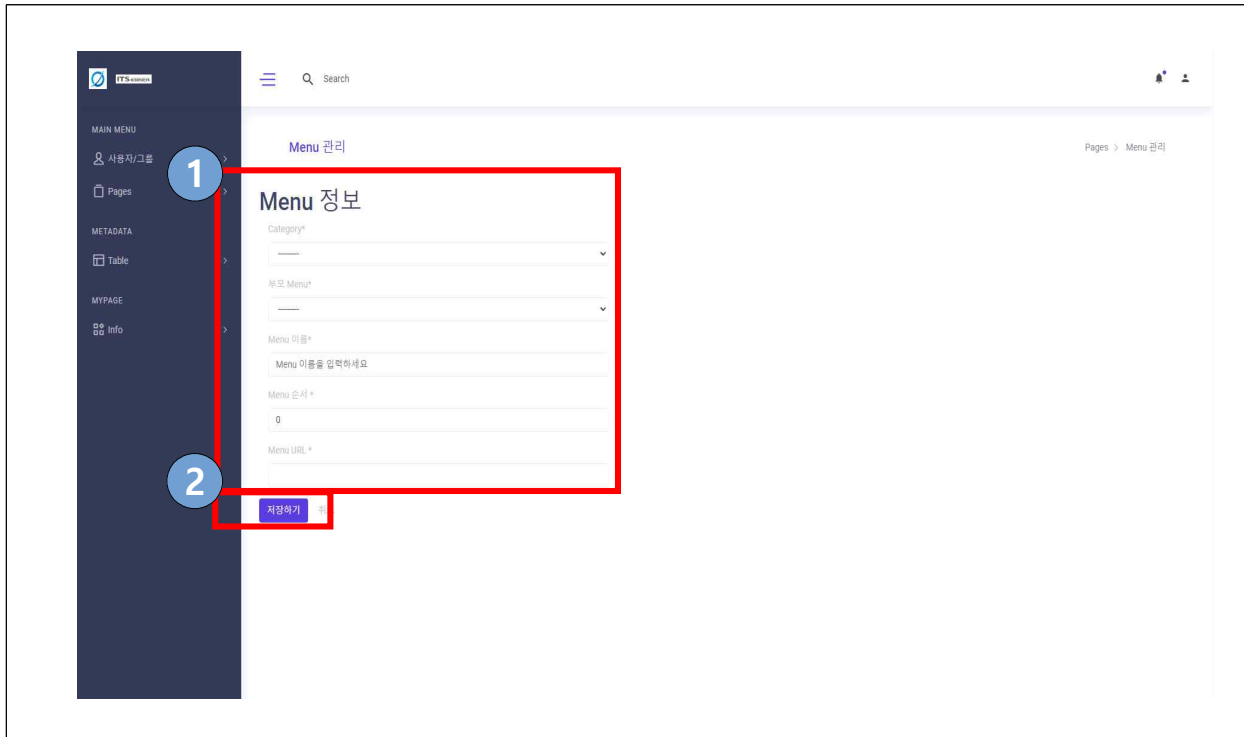
11.5.14. 메뉴 관리



화면 개요	메뉴 관리 화면	
화면 설명	1	메뉴 등록 버튼
	2	전체 메뉴 목록 조회
사용 방법	1. 메뉴 : Pages > Menu	
	2. 초기화면 - 메뉴 목록 조회	
특기 사항	3. 선택화면 - Menu 이름 선택 시 해당 메뉴 정보 조회 가능	
	없음	

(표.43) 메뉴 관리

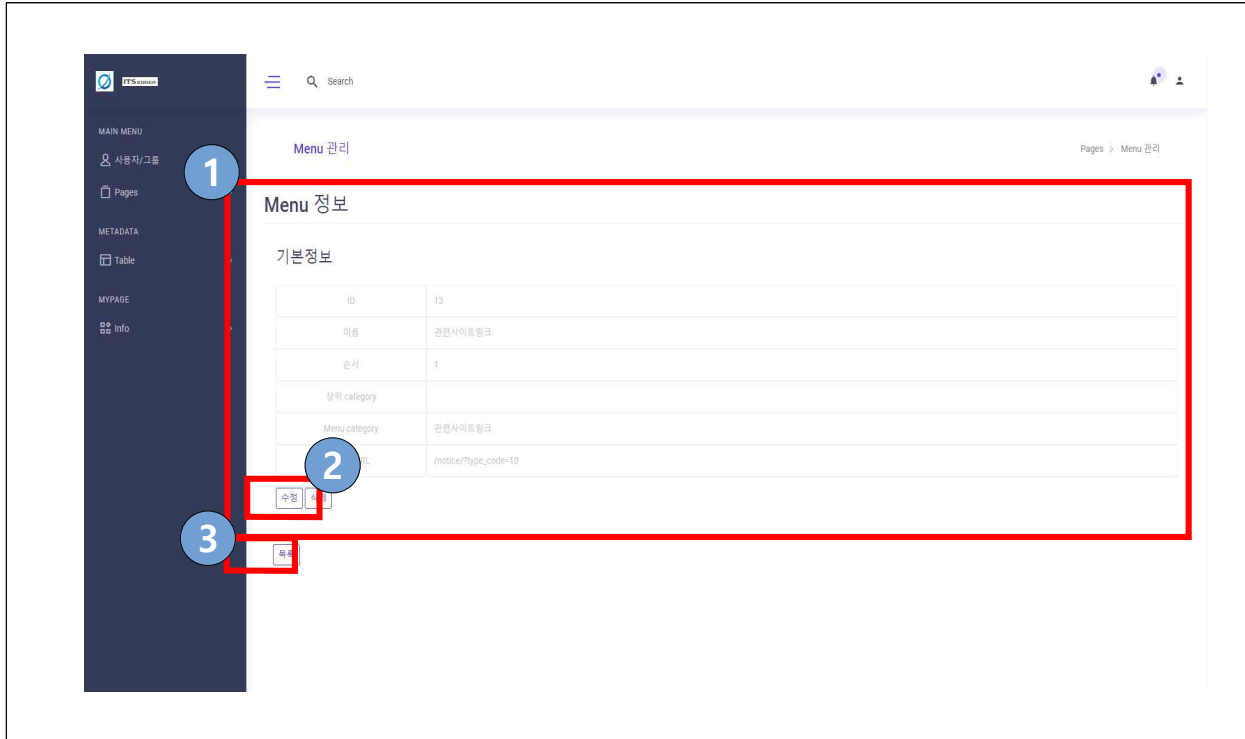
11.5.15. 메뉴 등록



화면 개요	메뉴 등록 화면	
화면 설명	1	메뉴 카테고리, 부모 메뉴 설정
	1	메뉴 이름 및 순서 입력
	1	메뉴 URL 설정
	2	'저장하기' 버튼
사용 방법	1. 메뉴 : 메뉴 > 'menu 등록하기' 버튼	
	2. 초기화면 - 카테고리, 부모 메뉴 설정 - 메뉴 이름 및 순서 입력 - 메뉴 URL 설정 (ex. /notice/?type_code=7)	
특기 사항	3. 선택화면 - 버튼 클릭 후 생성	
	Menu의 Category는 Menu의 종류 부모 Menu는 어느 Main menu 밑에 표시될 것인가를 나타내며 Menu 순서는 0~양수 값으로 값이 작을수록 먼저 표시됨(왼쪽 혹은 위에 표시) Menu URL은 Menu가 특정 Page나 Link를 나타내는 경우 해당 url을 입력하고 그 이외의 경우에는 입력하지 않음	

(표.44) 메뉴 등록

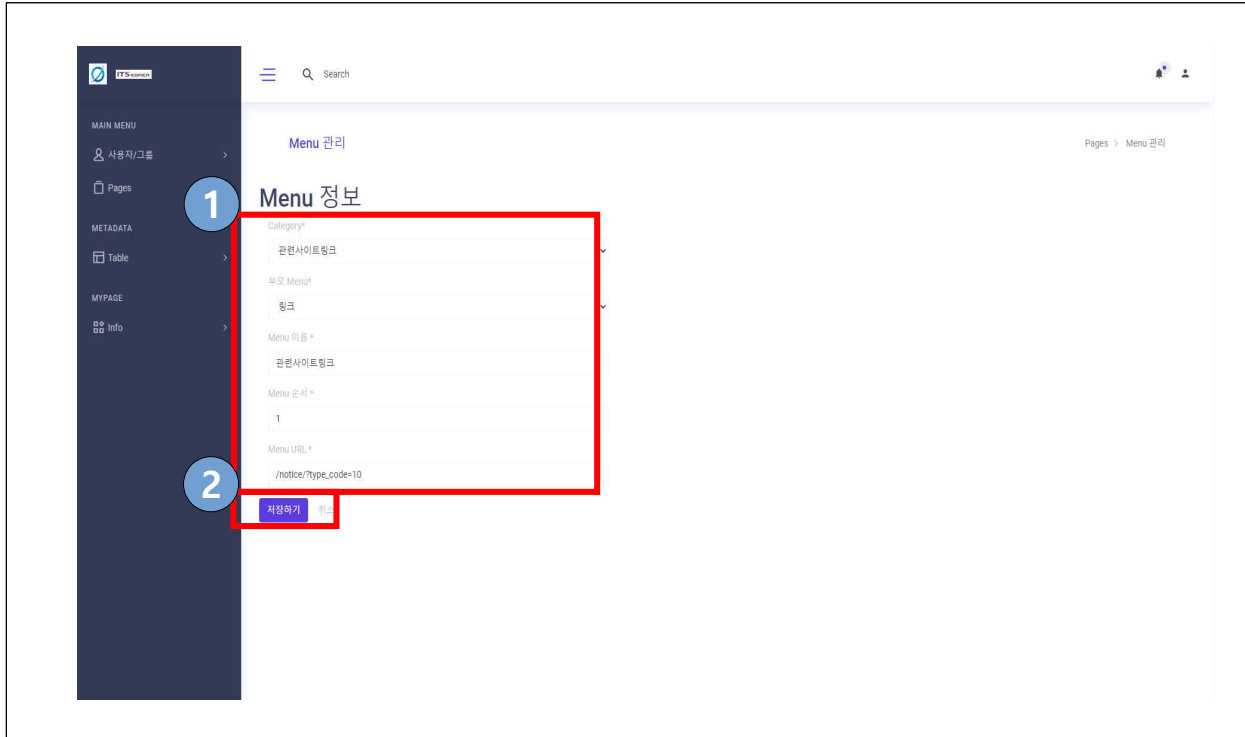
11.5.16. 메뉴 정보 조회



화면 개요	메뉴 정보 조회 화면	
화면 설명	1	메뉴의 기본 정보
	2	수정, 삭제 버튼
	3	'목록' 버튼
사용 방법	1. 메뉴 : 메뉴 목록 조회 > 메뉴 이름 선택 시 2. 초기화면 - 메뉴의 기본 정보 확인 가능 - 수정, 삭제 버튼을 통해 해당 메뉴를 수정하고 삭제 가능 - 목록 버튼 클릭 시 메뉴 목록 조회 페이지로 이동	
특기 사항	없음	

(표.45) 메뉴 정보 조회

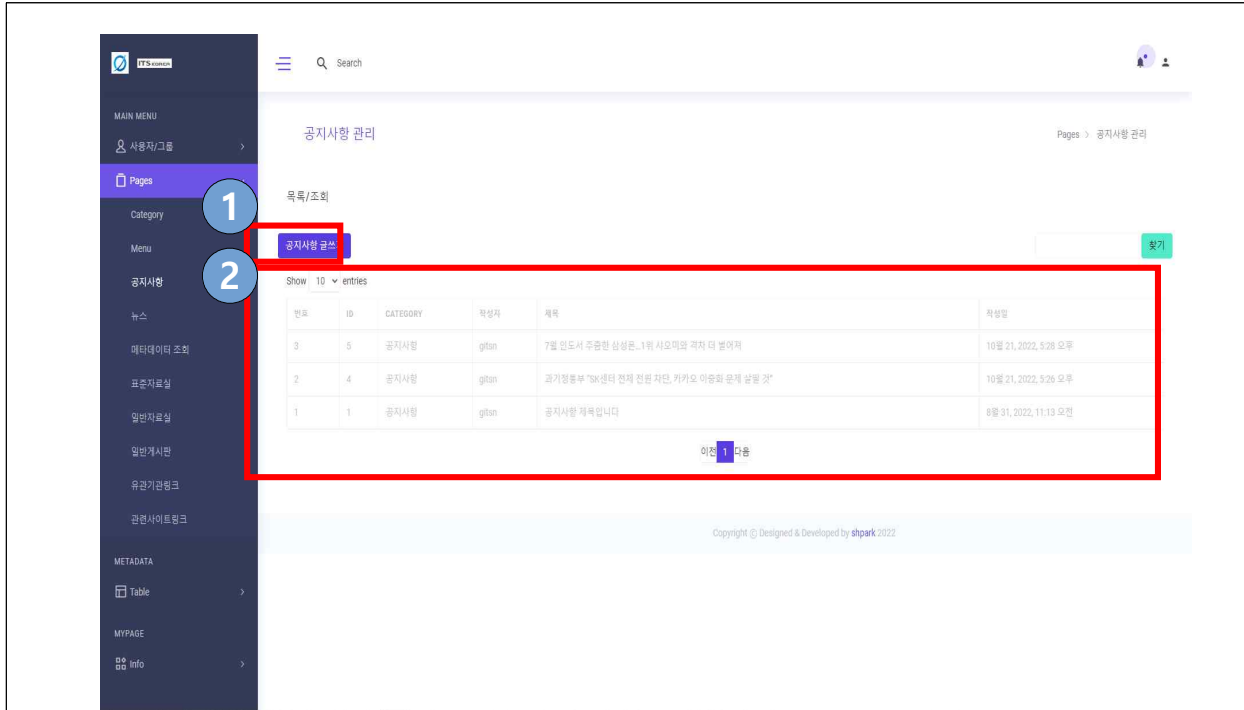
11.5.17. 메뉴 정보 수정



화면 개요	메뉴 정보 수정 화면	
화면 설명	1	등록되어있는 메뉴 기본 정보
	2	'저장하기' 버튼
사용 방법	1. 메뉴 : 메뉴 정보 > '수정' 버튼 2. 초기화면 - 메뉴의 기본 정보 수정 가능 3. 선택화면 - 버튼 클릭 후 저장	
특기 사항	Menu의 Category는 Menu의 종류 부모 Menu는 어느 Main menu 밑에 표시될 것인가를 나타내며 Menu 순서는 0~양수 값으로 값이 작을수록 먼저 표시됨(왼쪽 혹은 위에 표시) Menu URL은 Menu가 특정 Page나 Link를 나타내는 경우 해당 url을 입력하고 그 이외의 경우에는 자동 입력됨	

(표.46) 메뉴 정보 수정

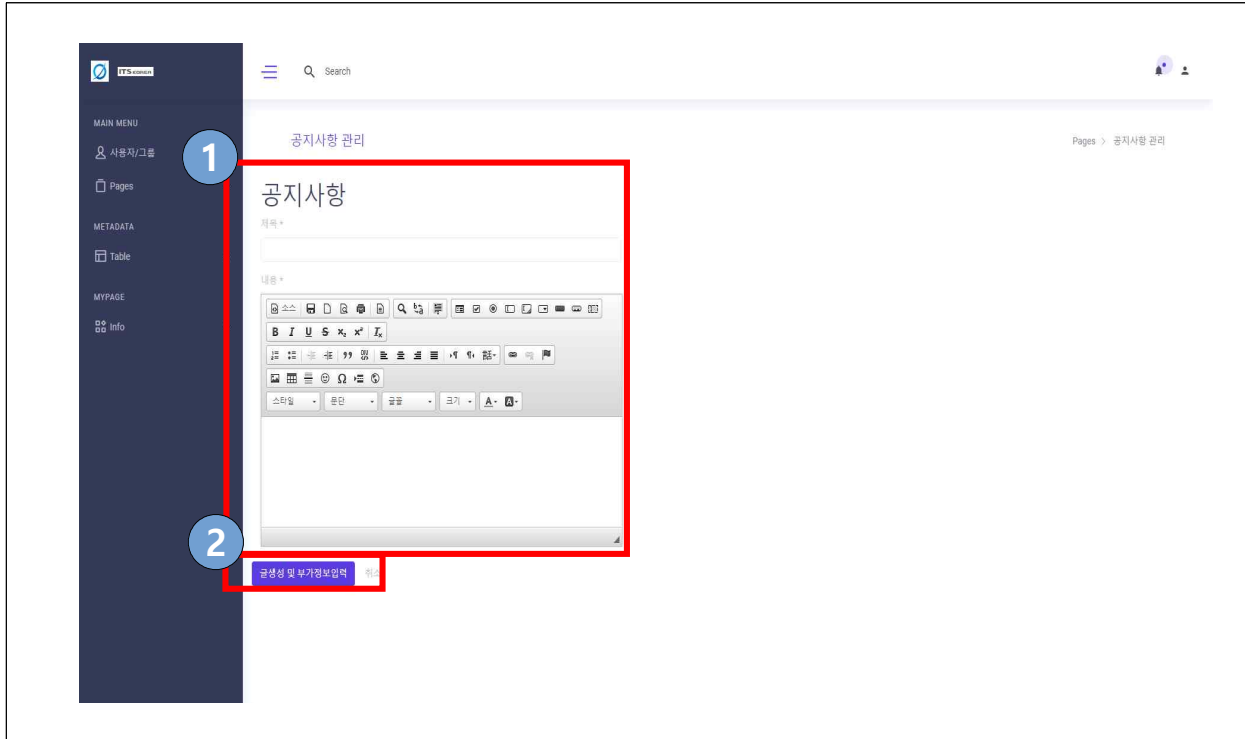
11.5.18. 공지사항 관리



화면 개요	공지사항 관리 화면	
화면 설명	1	공지사항 등록 버튼
	2	전체 공지사항 목록 조회
사용 방법	1. 메뉴 : Pages > 공지사항 2. 초기화면 - 공지사항 목록 조회 3. 선택화면 - 공지사항 제목 선택 시 해당 공지사항 조회 가능	
특기 사항	뉴스와 사용법 같음	

(표.47) 공지사항 관리

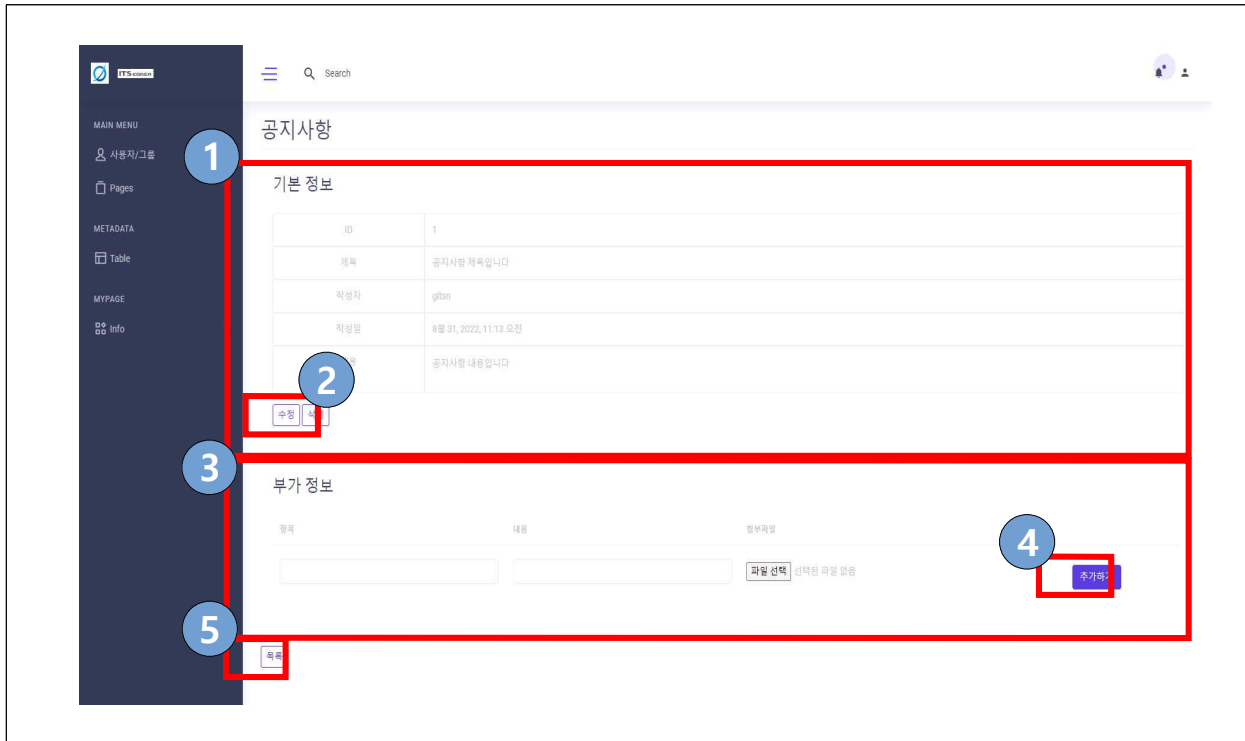
11.5.19. 공지사항 글쓰기



화면 개요	공지사항 등록 화면	
화면 설명	1	공지사항 제목 입력
	1	편집기에 내용 삽입
	2	'글 생성 및 부가정보 입력' 버튼
사용 방법	1. 메뉴 : 메뉴 > '공지사항 글쓰기' 버튼	
	2. 초기화면 - 공지사항 제목 입력 - 편집기에 공지사항 내용 삽입	
특기 사항	3. 선택화면 - '글생성 및 부가정보입력' 버튼 클릭 후 글 생성	
	뉴스와 사용법 같음	

(표.48) 공지사항 글쓰기

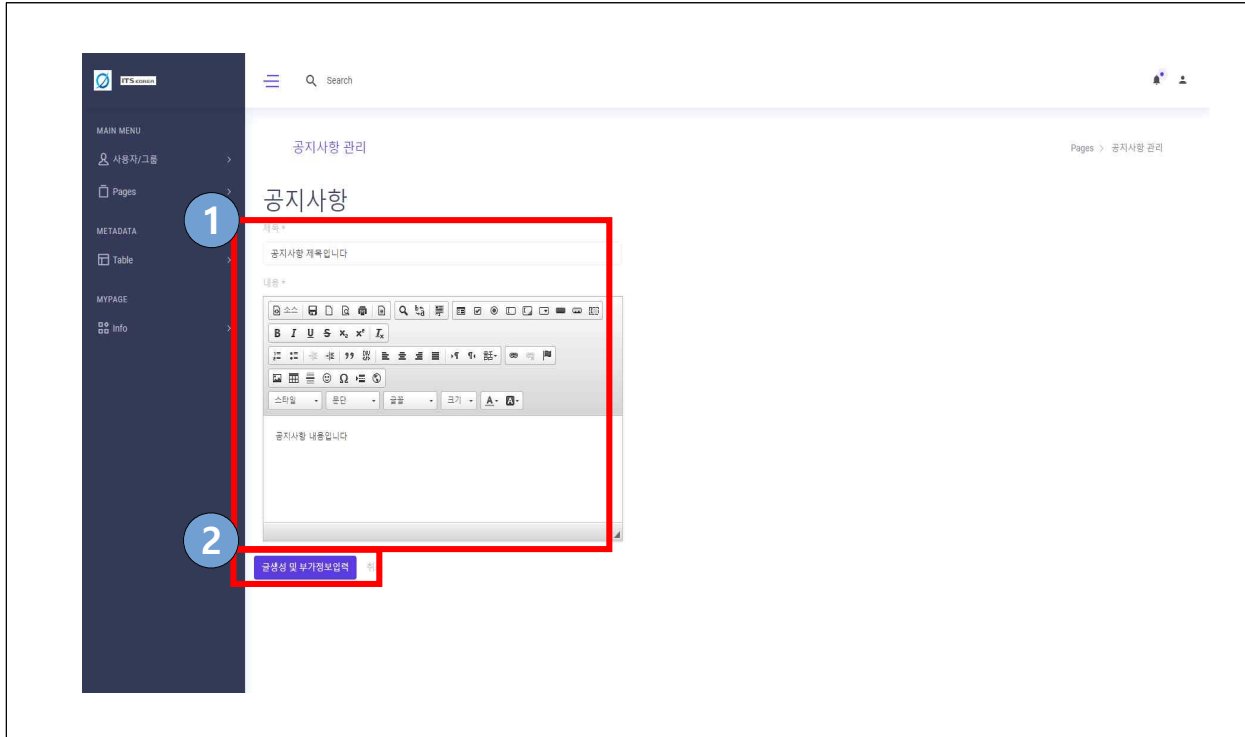
11.5.20. 공지사항 정보 조회



화면 개요	공지사항 정보 조회 화면	
화면 설명	1	공지사항의 기본 정보
	2	수정, 삭제 버튼
	3	부가정보 입력란
	4	부가정보 추가 버튼
	5	'목록' 버튼
사용 방법	1. 메뉴 : 공지사항 목록 조회 > 공지사항 정보	
	2. 초기화면	
	<ul style="list-style-type: none"> - 공지사항의 기본 정보 확인 가능 - 수정, 삭제 버튼을 통해 내용을 수정하고 삭제 가능 - 부가정보 항목, 내용, 첨부파일 추가 가능 - 목록 버튼 클릭 시 목록 조회 페이지로 이동 	
특기 사항	3. 선택화면	
	<ul style="list-style-type: none"> - 수정, 삭제를 클릭하여 내용 수정 및 삭제 가능 	
특기 사항	뉴스와 사용법 같음	

(표.49) 공지사항 정보 조회

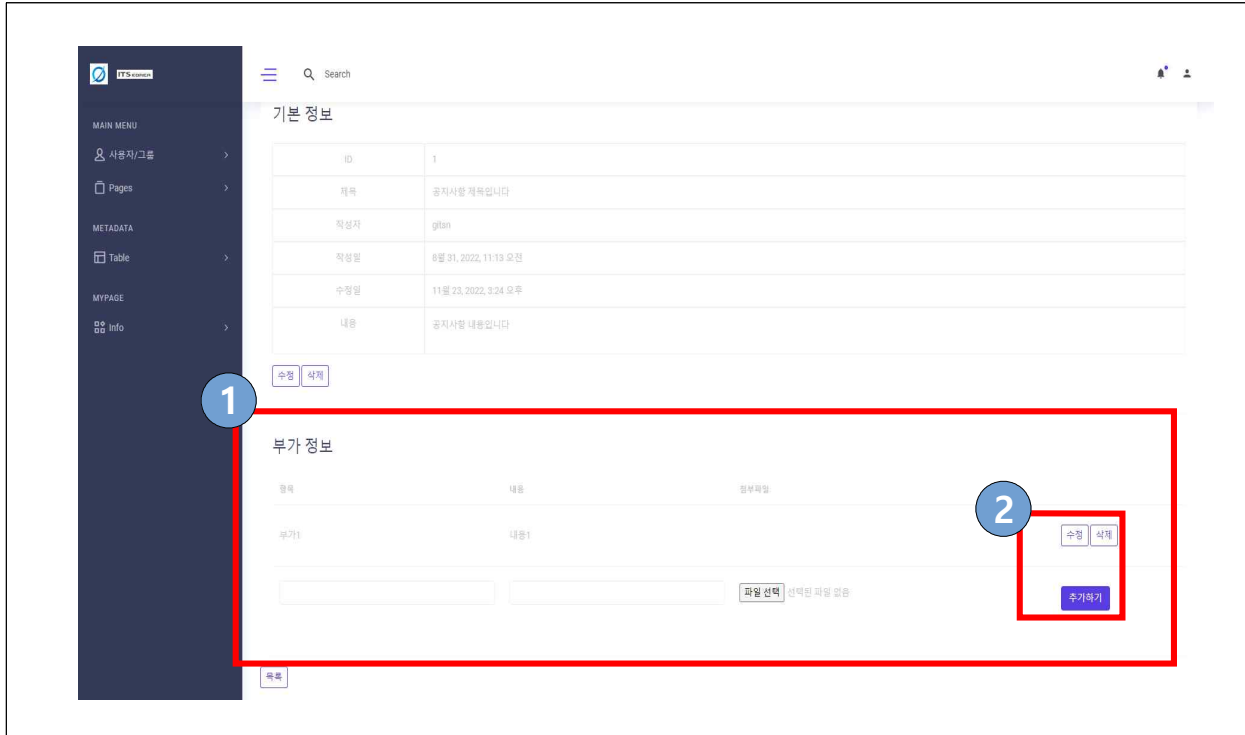
11.5.21. 공지사항 정보 수정



화면 개요	공지사항 정보 수정 화면	
화면 설명	1	등록되어있는 공지사항 기본 정보
	2	'글 생성 및 부가정보 입력' 버튼
사용 방법	1. 메뉴 : 공지사항 정보 > '수정' 버튼 2. 초기화면 - 공지사항의 기본 정보 수정 가능 3. 선택화면 - '글생성 및 부가정보입력' 버튼 클릭 후 내용 저장	
특기 사항	뉴스와 사용법 같음	

(표.50) 공지사항 정보 수정

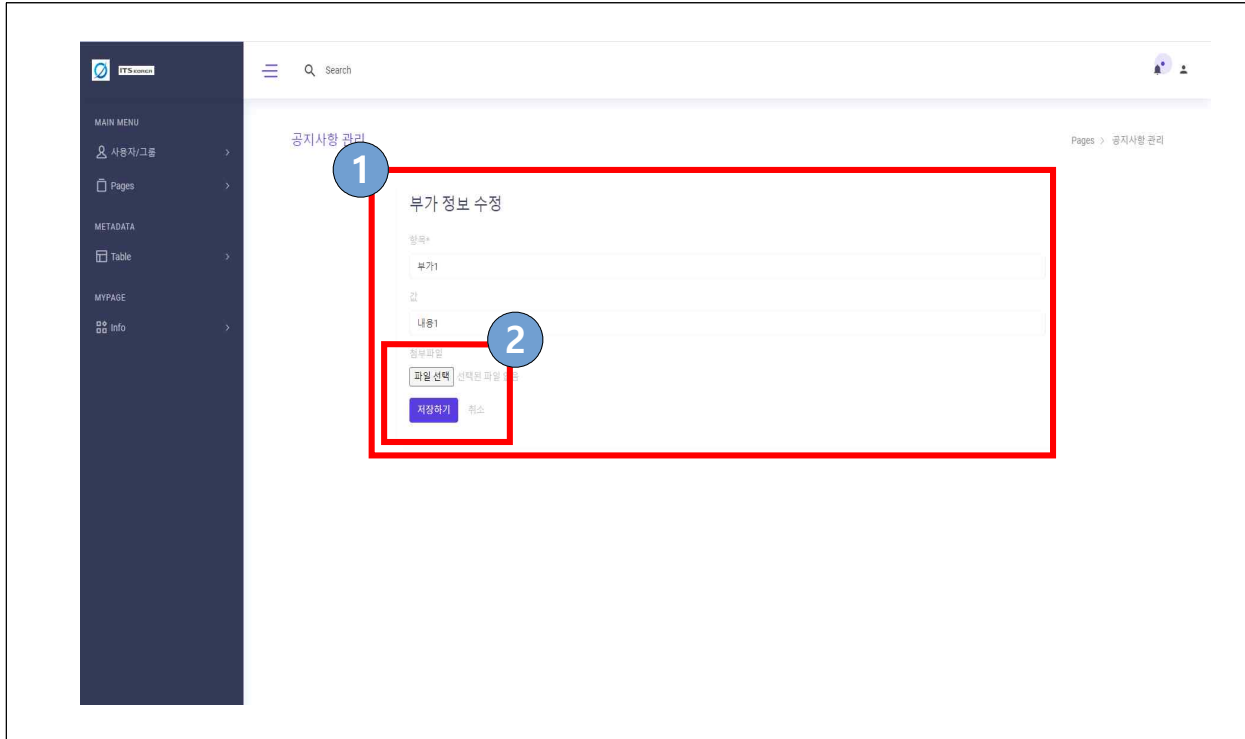
11.5.22. 공지사항 부가정보 추가, 삭제



화면 개요	공지사항 부가정보 추가, 삭제 화면	
화면 설명	1	등록되어있는 공지사항 부가정보
	2	추가, 수정, 삭제 버튼
사용 방법	1. 메뉴 : 공지사항 정보 > 부가정보 '추가' 버튼 2. 초기화면 - 공지사항의 부가정보 추가 및 삭제 가능 - 공지사항의 부가정보 수정 가능 3. 선택화면 - 수정, 삭제, 추가하기 버튼 클릭 후 부가 정보 수정, 삭제, 추가 가능	
특기 사항	뉴스와 사용법 같음	

(표.51) 공지사항 부가정보 추가,삭제

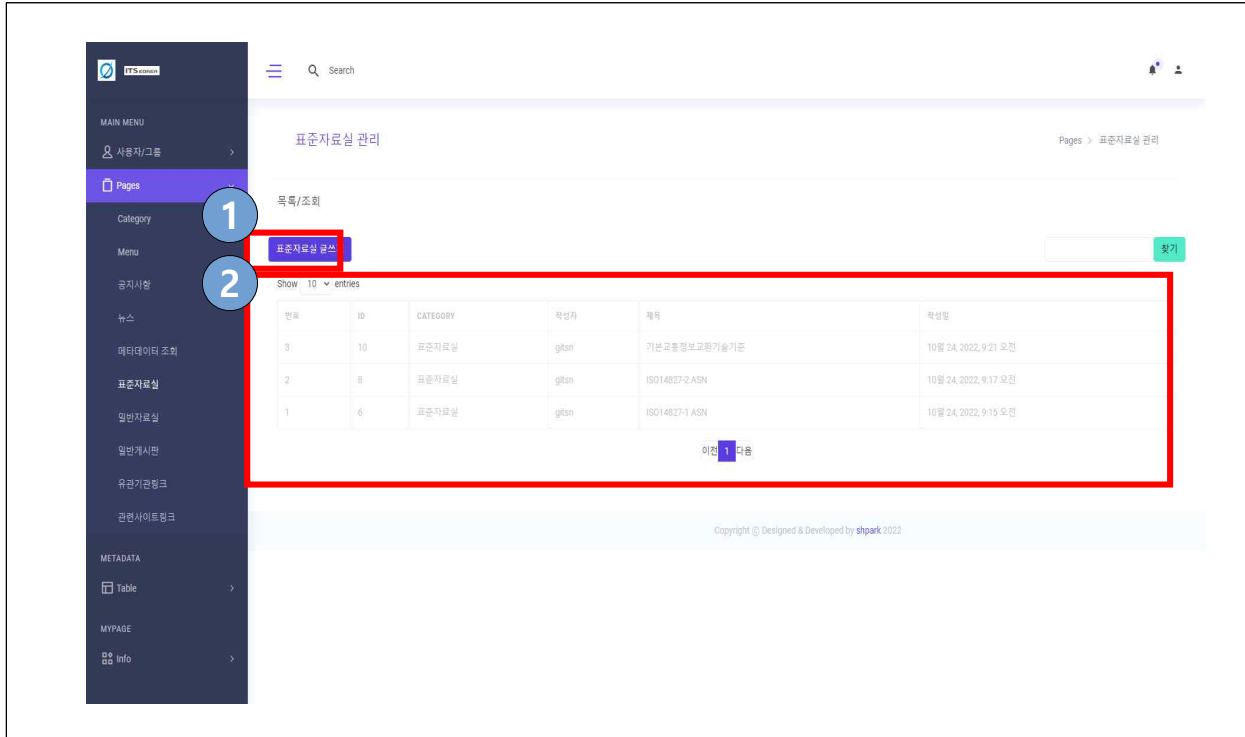
11.5.23. 공지사항 추가정보 수정



화면 개요	공지사항 추가정보 수정 화면	
화면 설명	1	등록되어있는 공지사항 추가정보 수정
	2	파일 첨부 버튼 및 저장 버튼
사용 방법	1. 메뉴 : 공지사항 추가정보 > 추가정보 '수정' 버튼 2. 초기화면 - 공지사항의 추가정보 수정 가능 - 파일 첨부 기능, 파일 삭제 기능 3. 선택화면 - 저장하기 버튼 클릭 후 저장 가능	
특기 사항	뉴스와 사용법 같음 파일이 첨부되어 있는 경우 '취소' 를 선택하면 첨부된 파일을 삭제 가능	

(표.52) 공지사항 추가정보 수정

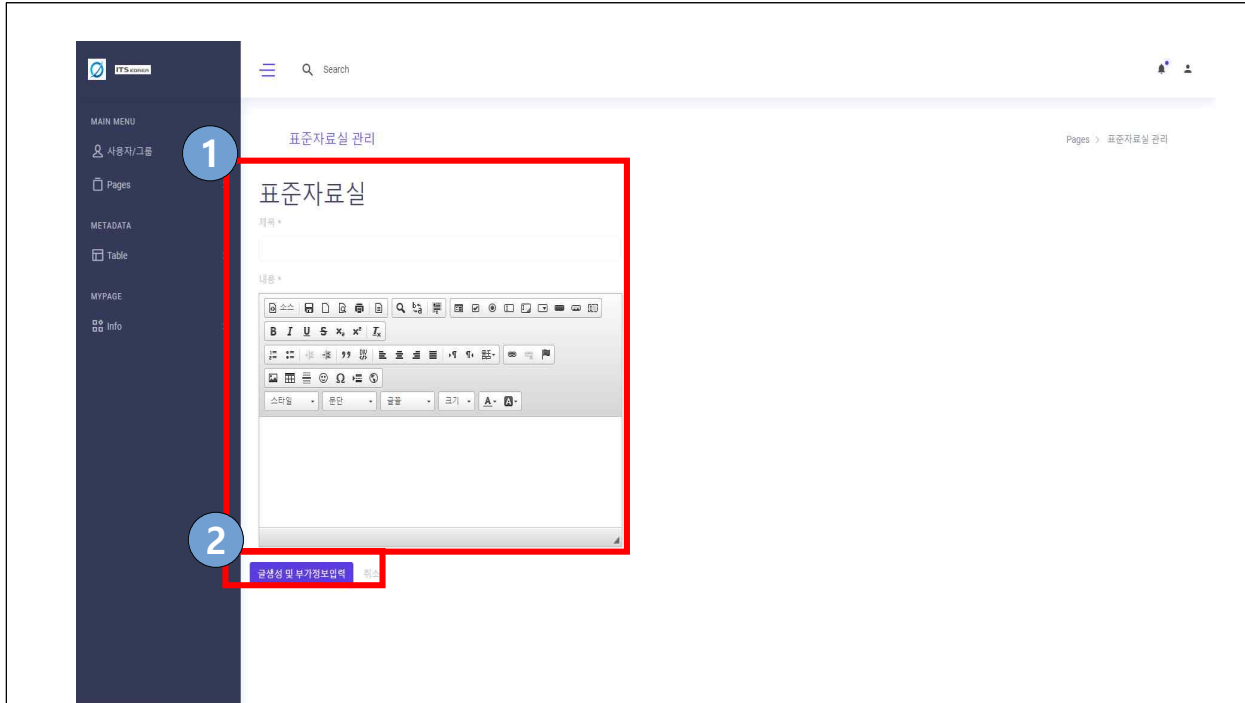
11.5.24. 표준자료실 관리



화면 개요	표준자료실 관리 화면	
화면 설명	1	표준자료 등록 버튼
	2	전체 표준자료 목록 조회
사용 방법	1. 메뉴 : Pages > 표준자료실 2. 초기화면 - 표준자료 목록 조회 3. 선택화면 - 제목 선택 시 해당 표준자료 조회 가능	
특기 사항	일반자료실과 사용법 같음	

(표.53) 표준자료실 관리

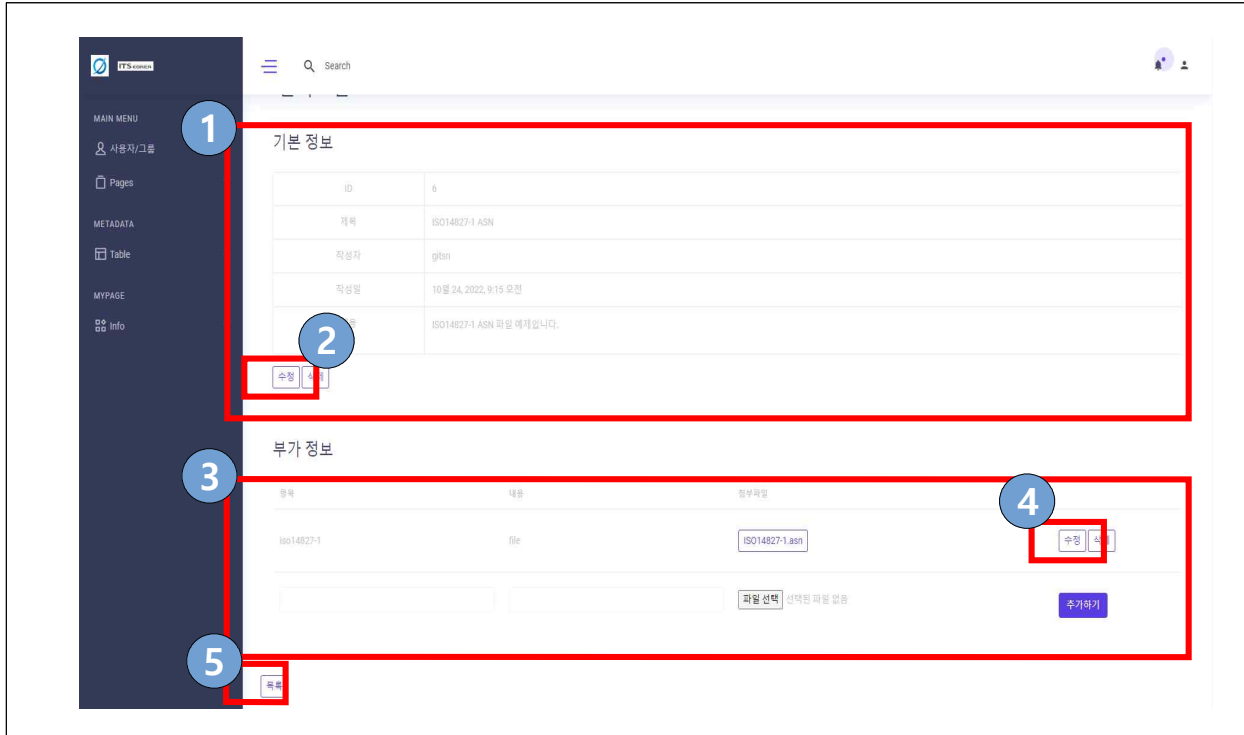
11.5.25. 표준자료실 글쓰기



화면 개요	표준자료실 등록 화면	
화면 설명	1	표준자료 제목 입력
	1	편집기에 내용 삽입
	2	'글 생성 및 추가정보 입력' 버튼
사용 방법	1. 메뉴 : 표준자료실 목록 > '표준자료실 글쓰기' 버튼	
	2. 초기화면	
	- 표준자료실 제목 입력 - 편집기에 표준자료실 내용 입력	
특기 사항	3. 선택화면	
	- '글생성 및 추가정보 입력' 버튼 클릭 후 저장 가능	
특기 사항	일반자료실과 사용법 같음	

(표.54) 표준자료실 글쓰기

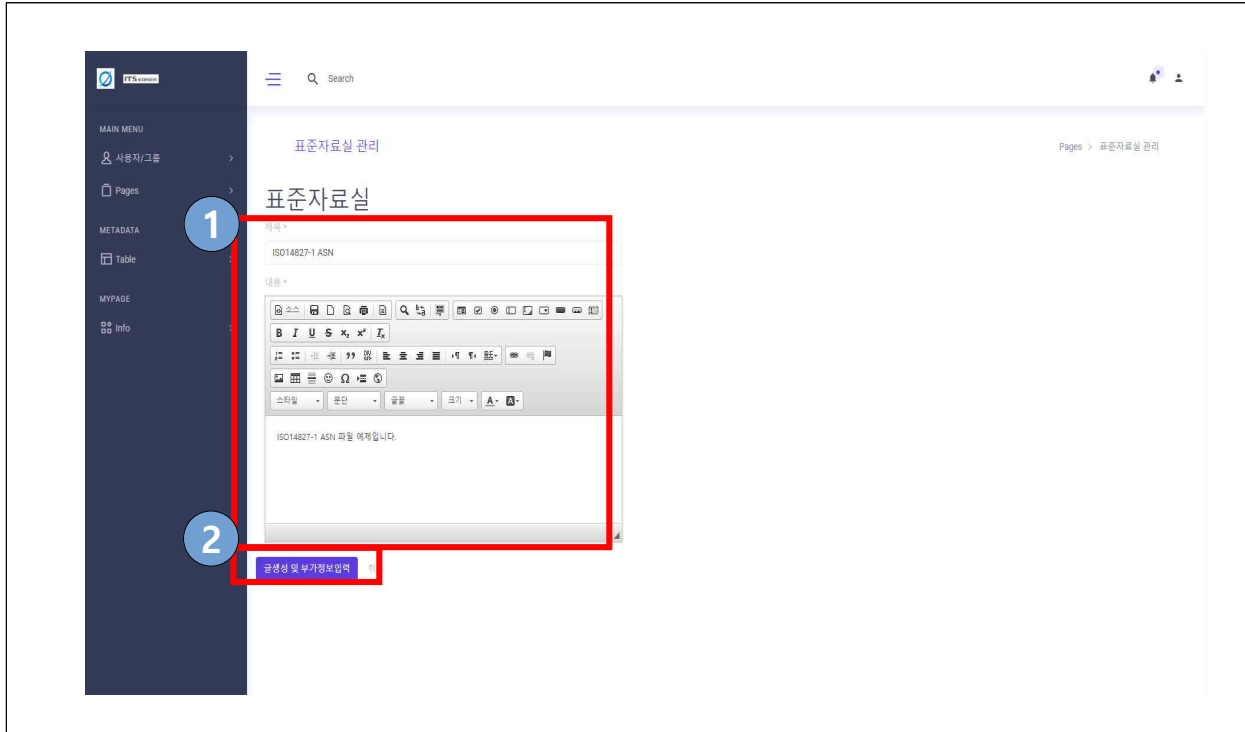
11.5.26. 표준자료실 정보 조회



화면 개요	표준자료실 정보 조회 화면	
화면 설명	1	표준자료의 기본 정보
	2	수정, 삭제 버튼
	3	부가정보 입력란
	4	부가정보 추가 버튼
	5	'목록' 버튼
사용 방법	1. 메뉴 : 표준자료실 목록 조회 > 표준자료 정보	
	2. 초기화면 - 표준자료의 기본 정보 확인 가능 - 수정, 삭제 버튼을 통해 수정하고 삭제 가능 - 부가정보 항목, 내용, 첨부파일 첨부 후 추가 가능 - 목록 버튼 클릭 시 표준자료실 목록 조회 페이지로 이동	
특기 사항	3. 선택화면 - 수정, 삭제, 추가하기 버튼 클릭 후 부가 정보 수정, 삭제, 추가 가능	
	일반자료실과 사용법 같음	

(표.55) 표준자료실 정보 조회

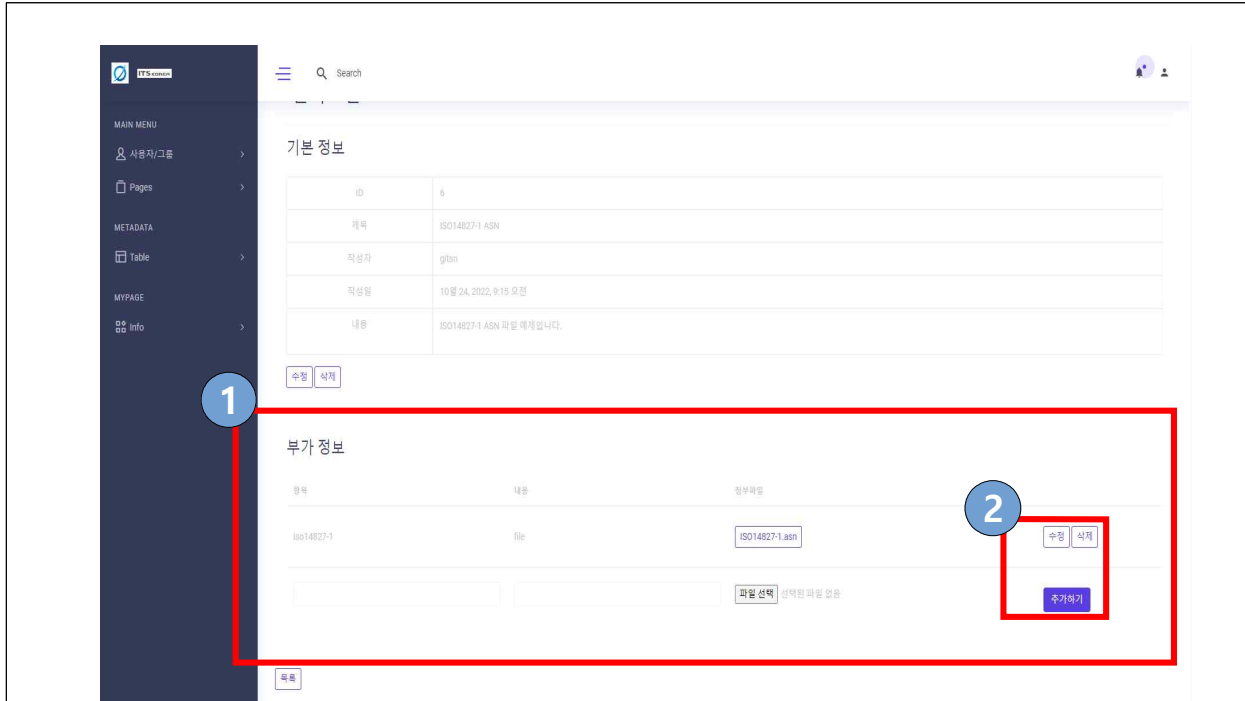
11.5.27. 표준자료실 정보 수정



화면 개요	표준자료실 정보 수정 화면	
화면 설명	1	등록되어있는 표준자료 기본 정보
	2	‘글 생성 및 부가정보 입력’ 버튼
사용 방법	1. 메뉴 : 표준자료실 정보 > ‘수정’ 버튼 2. 초기화면 - 표준자료실의 기본 정보 수정 가능 3. 선택화면 - ‘글 생성 및 부가정보 입력’ 버튼 클릭 후 저장	
특기 사항	일반자료실과 사용법 같음	

(표.56) 표준자료실 정보 수정

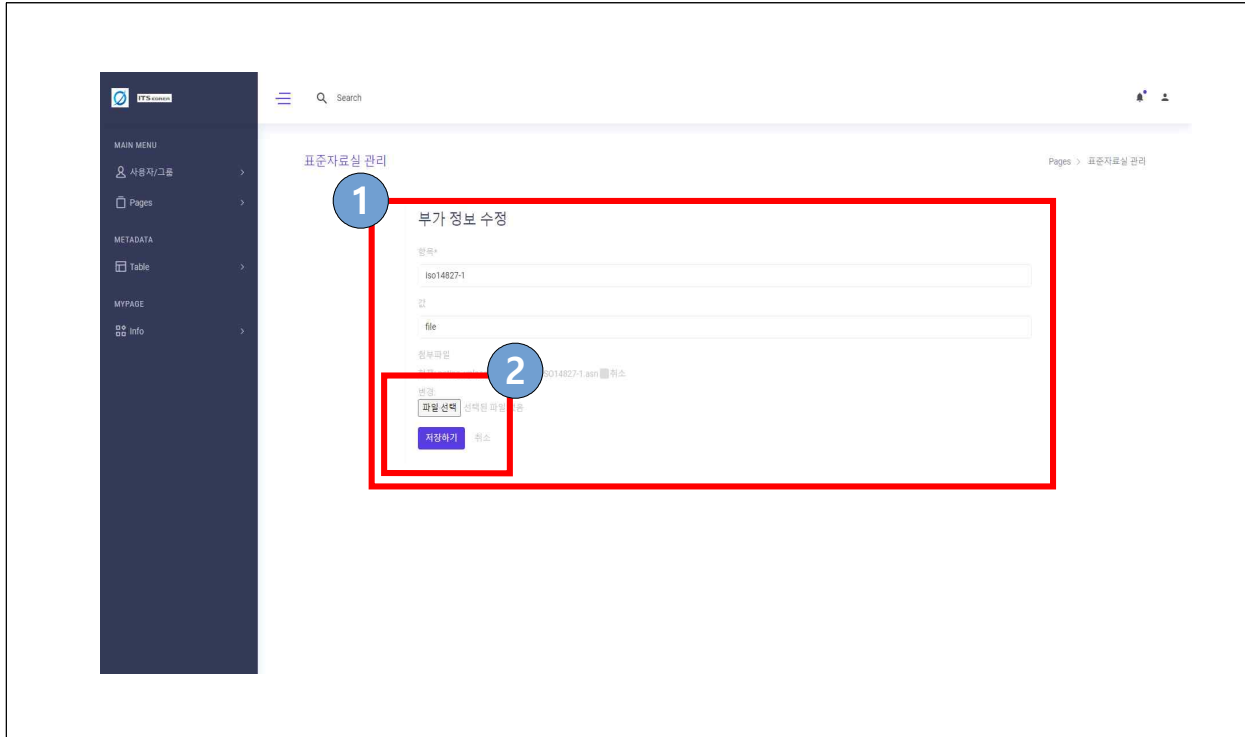
11.5.28. 표준자료실 부가정보 추가, 삭제



화면 개요	표준자료실 부가정보 추가, 삭제 화면	
화면 설명	1	등록되어있는 표준자료 부가정보
	2	추가, 수정, 삭제 버튼
사용 방법	1. 메뉴 : 표준자료실 정보 > 부가정보 '추가' 버튼 2. 초기화면 - 표준자료실의 부가정보 추가 및 삭제 가능 3. 선택화면 - 수정, 삭제, 추가하기 버튼 클릭 후 부가 정보 수정, 삭제, 추가 가능	
특기 사항	일반자료실과 사용법 같음	

(표.57) 표준자료실 부가정보 추가,삭제

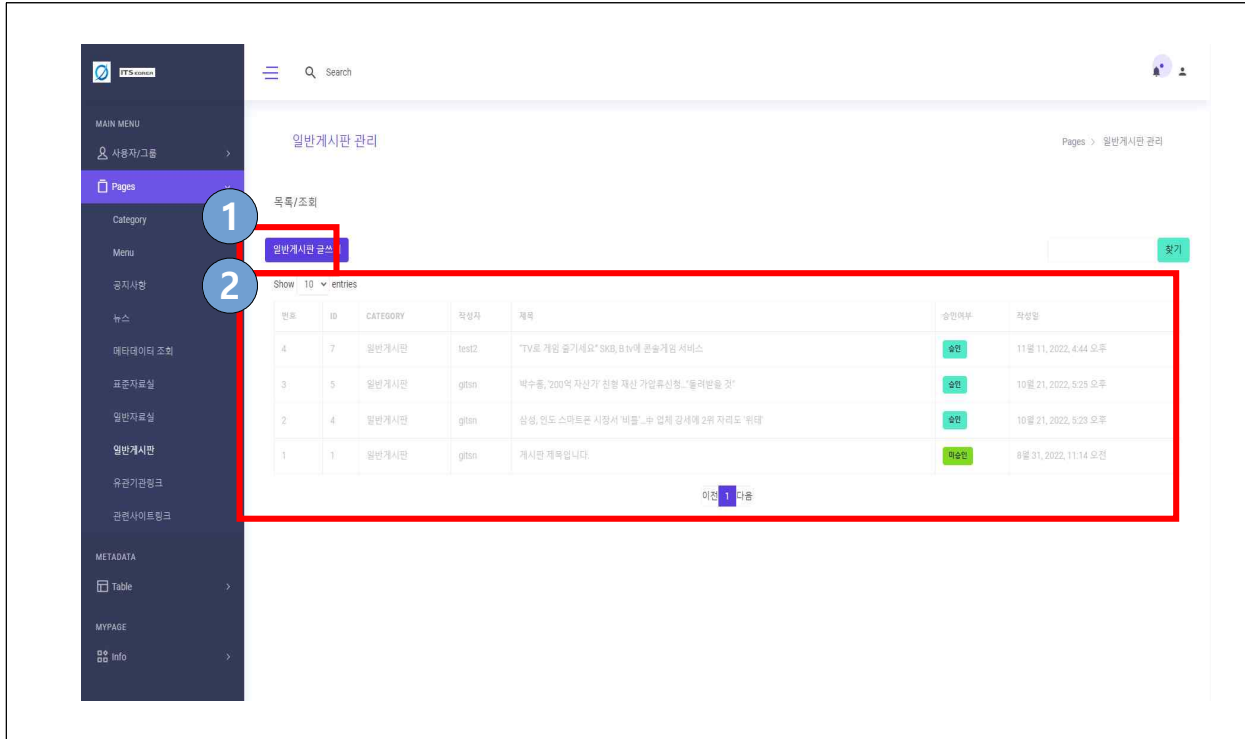
11.5.29. 표준자료실 부가정보 수정



화면 개요	표준자료실 부가정보 수정 화면	
화면 설명	1	등록되어있는 표준자료 부가정보 수정
	2	파일 첨부 버튼 및 저장 버튼
사용 방법	<ol style="list-style-type: none"> 1. 메뉴 : 표준자료실 부가정보 > 부가정보 '수정' 버튼 2. 초기화면 <ul style="list-style-type: none"> - 표준자료실의 부가정보 수정 가능 - 파일 첨부 기능, 파일 삭제 기능 3. 선택화면 <ul style="list-style-type: none"> - 저장하기 버튼 클릭 후 저장 가능 	
특기 사항	<p>일반자료실과 사용법 같음</p> <p>파일이 첨부되어 있는 경우 '취소' 를 선택하면 첨부된 파일을 삭제 가능</p>	

(표.58) 표준자료실 부가정보 수정

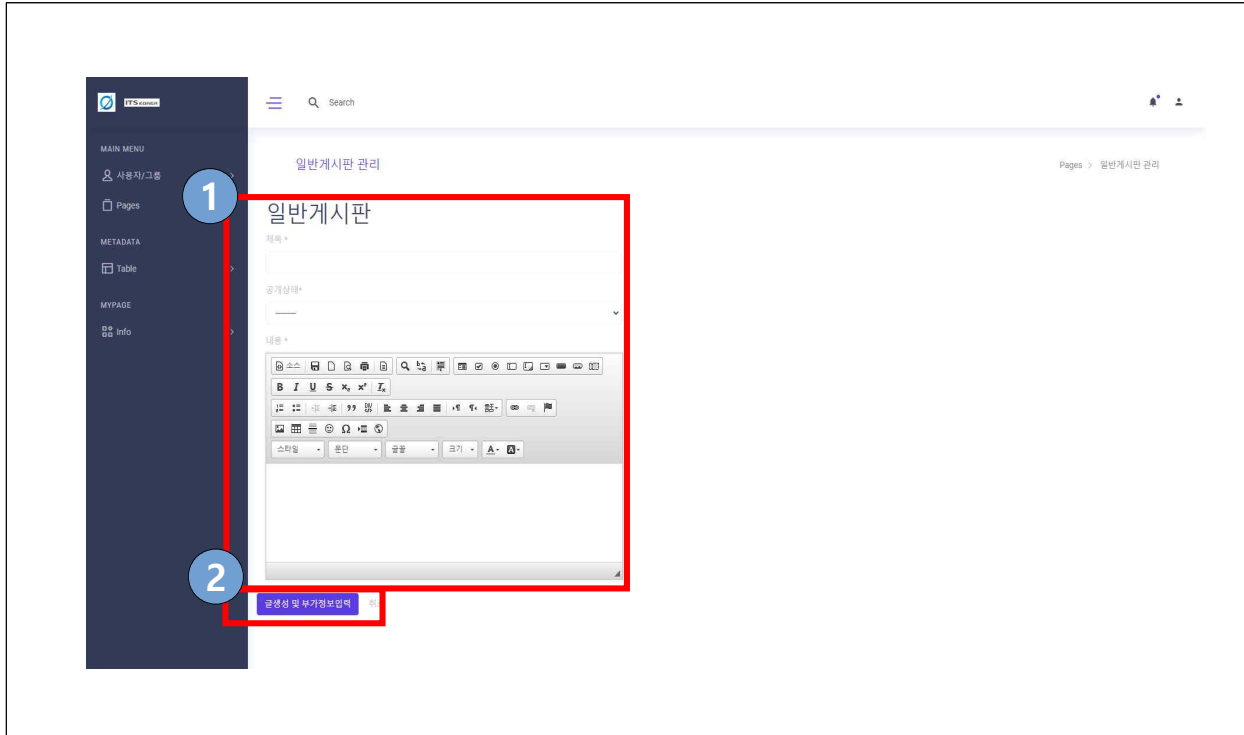
11.5.30. 일반게시판 관리



화면 개요	일반게시판 관리 화면	
화면 설명	1	일반게시판 등록 버튼
	2	전체 일반게시판 목록 조회
사용 방법	1. 메뉴 : Pages > 일반게시판 2. 초기화면 일반게시판 목록 조회 3. 선택화면 - 제목 선택 시 일반게시판 정보 조회 가능	
특기 사항	승인여부가 미승인 상태인 글의 경우 일반 사용자 페이지에서 표시되지 않음.	

(표.59) 일반게시판 관리

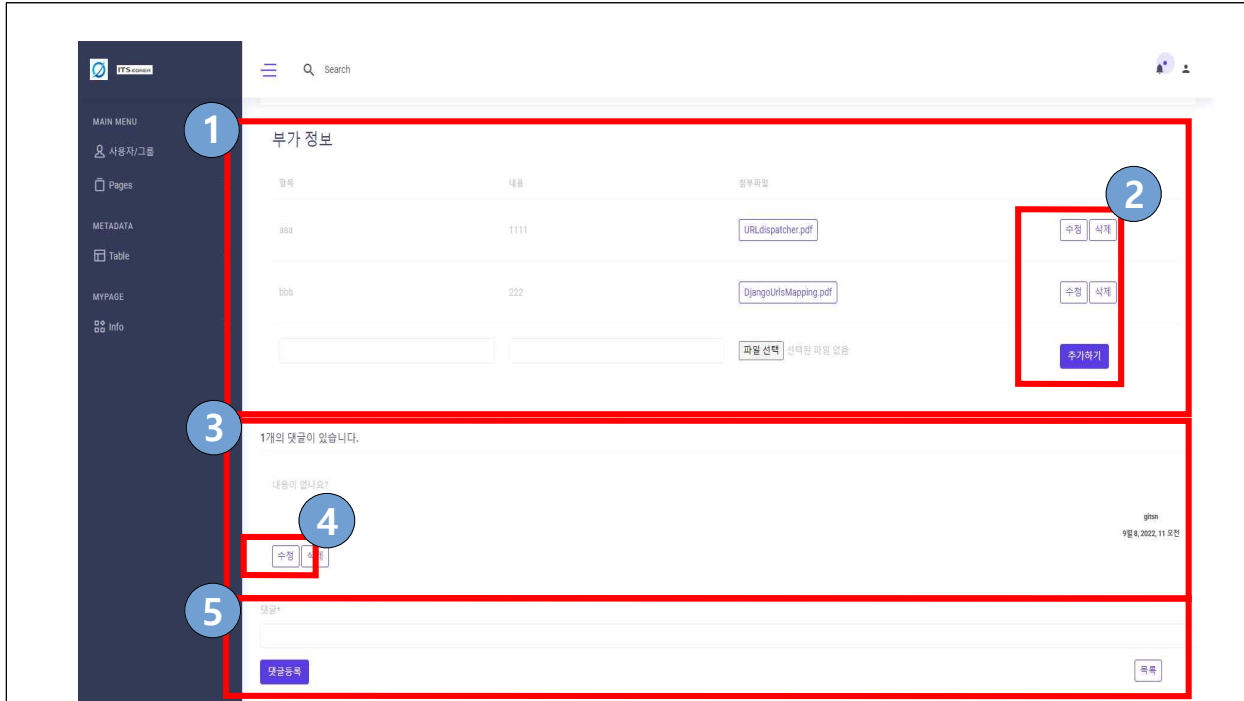
11.5.31. 일반게시판 글쓰기



화면 개요	일반게시판 등록 화면	
화면 설명	1	일반게시판 제목 입력
	1	편집기에 내용 삽입
	2	'글 생성 및 추가정보 입력' 버튼
사용 방법	1. 메뉴 : 일반게시판 목록 > '일반게시판 글쓰기' 버튼	
	2. 초기화면 - 일반게시판 제목 입력 - 편집기에 일반게시판 내용 삽입	
특기 사항	3. 선택화면 - '글생성 및 추가정보입력' 버튼 클릭 후 생성	
	편집기에 다른 웹브라우저의 화면을 복사해서 붙여넣기 해도 다른 웹 브라우저의 표시 화면과 동일한 내용 입력 가능	

(표.60) 일반게시판 글쓰기

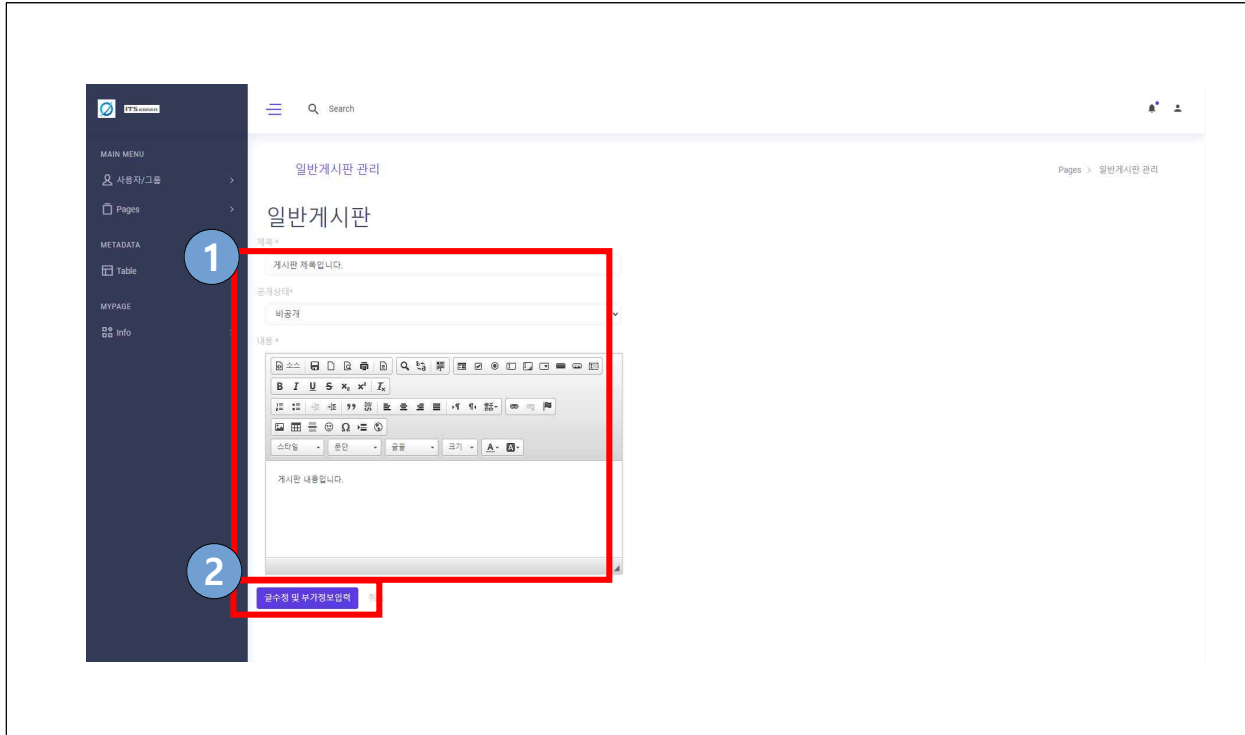
11.5.32. 일반게시판 정보 조회



화면 개요	일반게시판 정보 조회 화면	
화면 설명	1	부가정보 입력란
	2	부가정보 추가, 수정, 삭제 버튼
	3	관리자 댓글
	4	댓글 수정, 삭제 버튼
	5	댓글 기재 후 등록
사용 방법	1. 메뉴 : 일반게시판 목록 조회 > 일반게시판 정보	
	2. 초기화면 - 일반게시판의 상단에 기본 정보 확인 가능 - 수정, 삭제 버튼을 통해 수정하고 삭제 가능 - 부가정보 항목, 내용, 첨부파일 첨부 후 추가 가능 - 관리자 댓글 등록 및 수정, 삭제 가능 - 목록 버튼 클릭 시 목록 조회 페이지로 이동	
특기 사항	3. 선택화면 - 부가정보의 수정, 삭제를 클릭하여 부가정보의 내용 수정 및 삭제 가능	
	관리자가 해당 게시글에 댓글을 추가할 있음.	

(표.61) 일반게시판 정보 조회

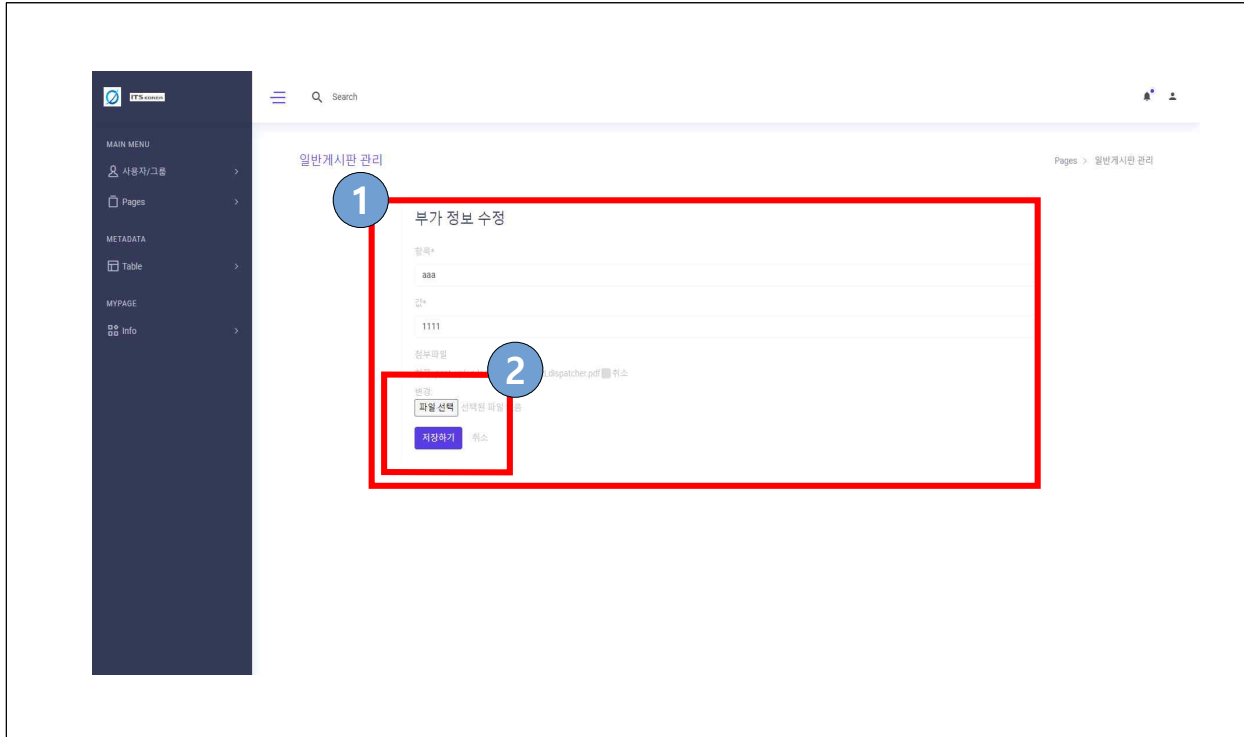
11.5.33. 일반게시판 정보 수정



화면 개요	일반게시판 정보 수정 화면	
화면 설명	1	등록되어있는 일반게시판 기본 정보
	2	'글 생성 및 부가정보입력' 버튼
사용 방법	1. 메뉴 : 일반게시판 정보 > '수정' 버튼 2. 초기화면 - 일반게시판의 기본 정보 수정 가능 3. 선택화면 - '글수정 및 부가정보입력' 버튼 클릭 후 저장	
특기 사항	없음	

(표.62) 일반게시판 정보 수정

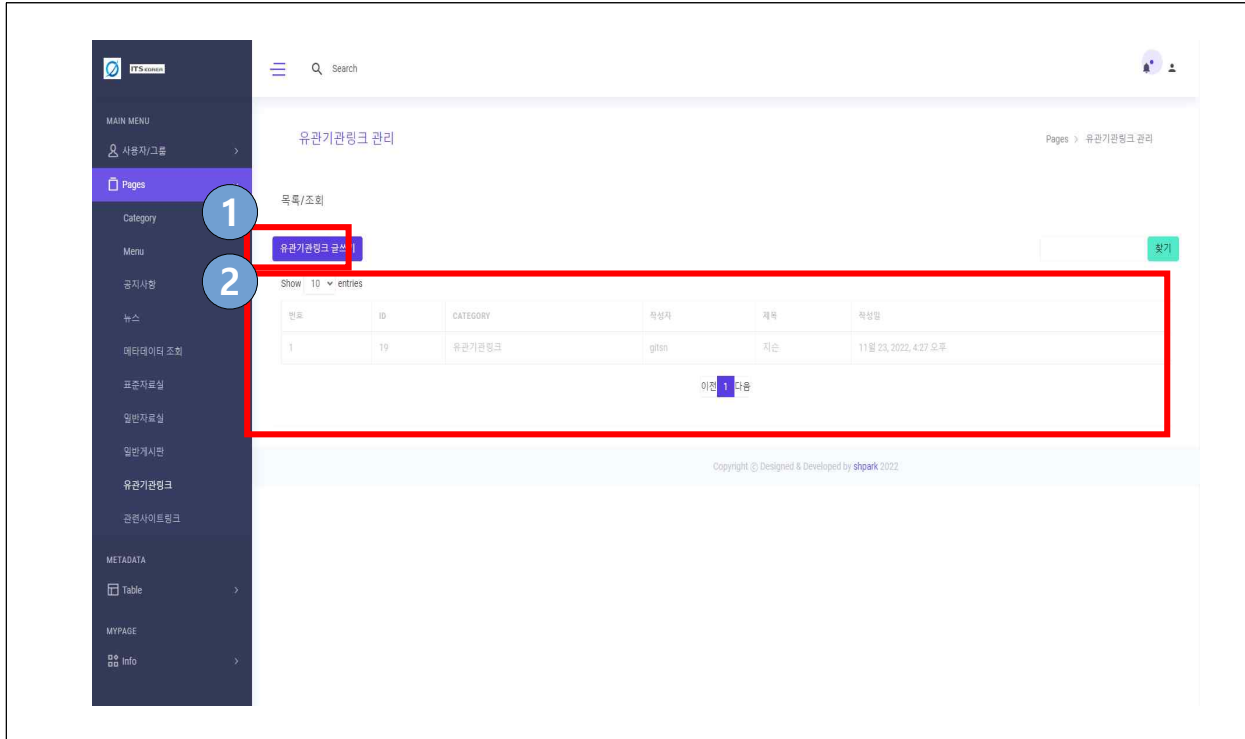
11.5.34. 일반게시판 추가정보 수정



화면 개요	일반게시판 추가정보 수정 화면	
화면 설명	1	등록되어있는 일반게시판 추가정보 수정
	2	파일 첨부 버튼 및 저장 버튼
사용 방법	1. 메뉴 : 일반게시판 추가정보 > 추가정보 '수정' 버튼 2. 초기화면 - 일반게시판의 추가정보 수정 가능 - 파일 첨부 기능, 파일 삭제 기능 3. 선택화면 - 저장하기 버튼 클릭 후 추가정보 수정 가능	
특기 사항	파일이 첨부되어 있는 경우 '취소' 를 선택하면 첨부된 파일을 삭제 가능	

(표.63) 일반게시판 추가정보 수정

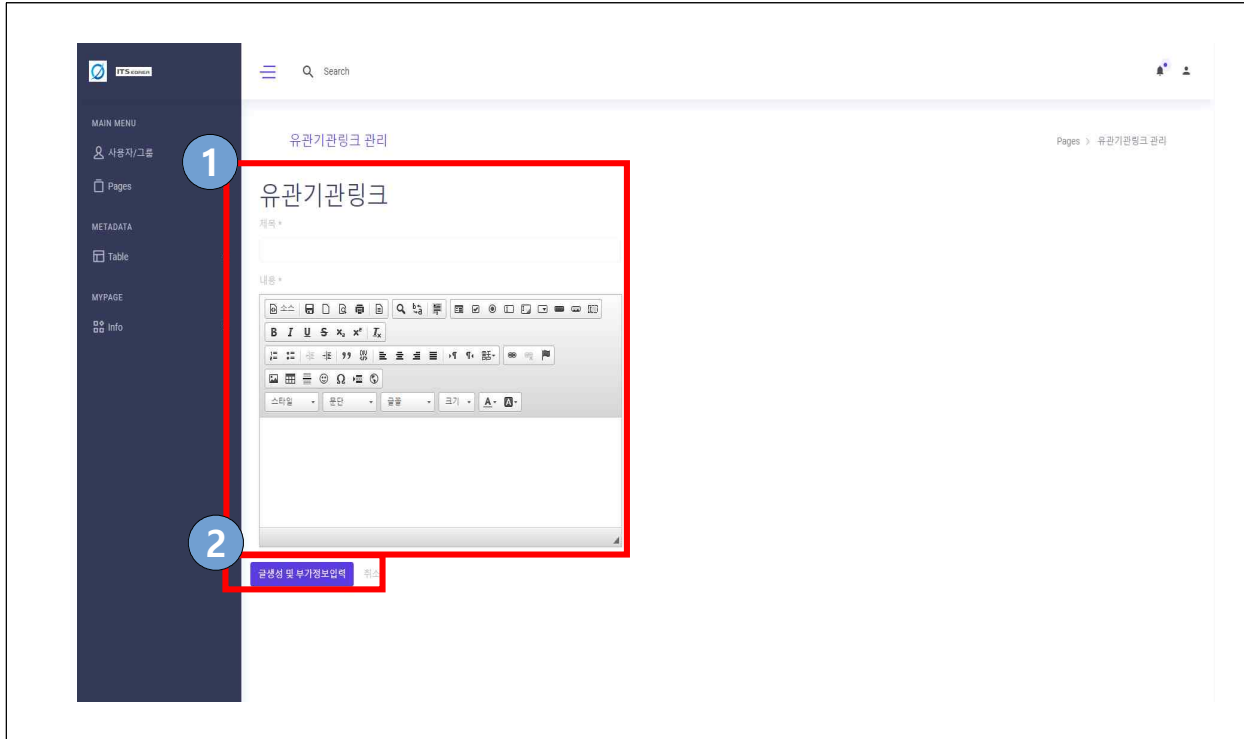
11.5.35. 유관기관 링크 관리




화면 개요	유관기관 링크 관리 화면	
화면 설명	1	유관기관 링크 등록 버튼
	2	유관기관 링크 목록 조회
사용 방법	1. 메뉴 : Pages > 유관기관 링크 2. 초기화면 - 유관기관 링크 목록 조회 3. 선택화면 - 제목 선택 시 해당 유관기관 링크 정보 조회 가능	
특기 사항	관련사이트 링크와 사용법 같음	

(표.64) 유관기관 링크 관리

11.5.36. 유관기관 링크 글쓰기



화면 개요	유관기관 링크 등록 화면
화면 설명	1 유관기관 링크 제목 입력
	1 편집기에 내용 삽입
	2 '글 생성 및 부가정보 입력' 버튼
사용 방법	1. 메뉴 : 유관기관 링크 목록 > '유관기관 링크 글쓰기' 버튼
	2. 초기화면
	- 유관기관 링크 제목 입력 - 편집기에 유관기관 링크 내용 삽입
특기 사항	3. 선택화면
	- '글생성 및 부가정보 입력' 버튼 클릭 후 생성
	편집기의 링크 버튼()을 통해서 보이는 글자와 실제 url 링크를 다르게 입력할 수 있음. 관련사이트 링크와 사용법 같음

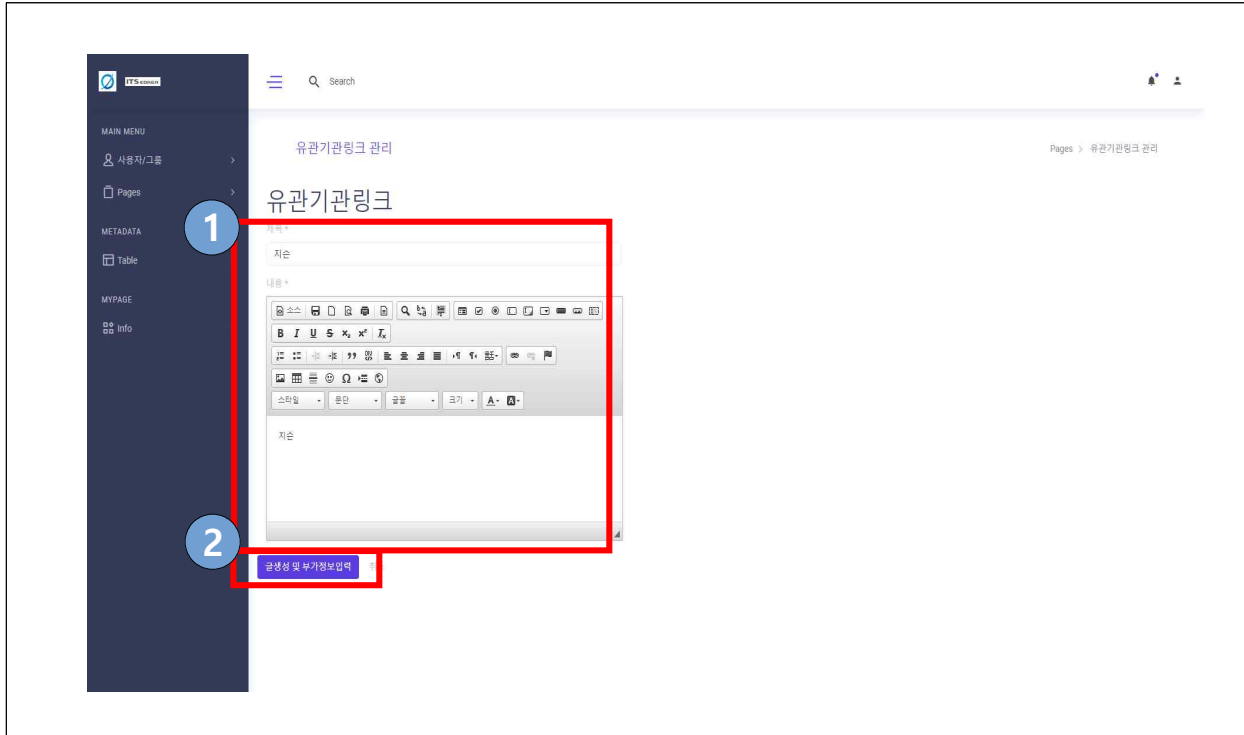
(표.65) 유관기관 링크 글쓰기


11.5.37. 유관기관 링크 정보 조회

화면 개요	유관기관 링크 정보 조회 화면	
화면 설명	1	유관기관 링크의 기본 정보
	2	수정, 삭제 버튼
	3	부가정보 입력란
	4	부가정보 추가 버튼
	5	'목록' 버튼
사용 방법	1. 메뉴 : 유관기관 링크 목록 조회 > 유관기관 링크 정보 2. 초기화면 - 유관기관 링크의 기본 정보 확인 가능 - 수정, 삭제 버튼을 통해 글 수정하고 삭제 가능 - 부가정보 항목, 내용, 첨부파일 추가 가능 - 목록 버튼 클릭 시 목록 조회 페이지로 이동 3. 선택화면 - 수정, 삭제를 클릭하여 해당 내용 수정 및 삭제 가능	
특기 사항	관련사이트 링크와 사용법 같음	

(표.66) 유관기관 링크 정보 조회

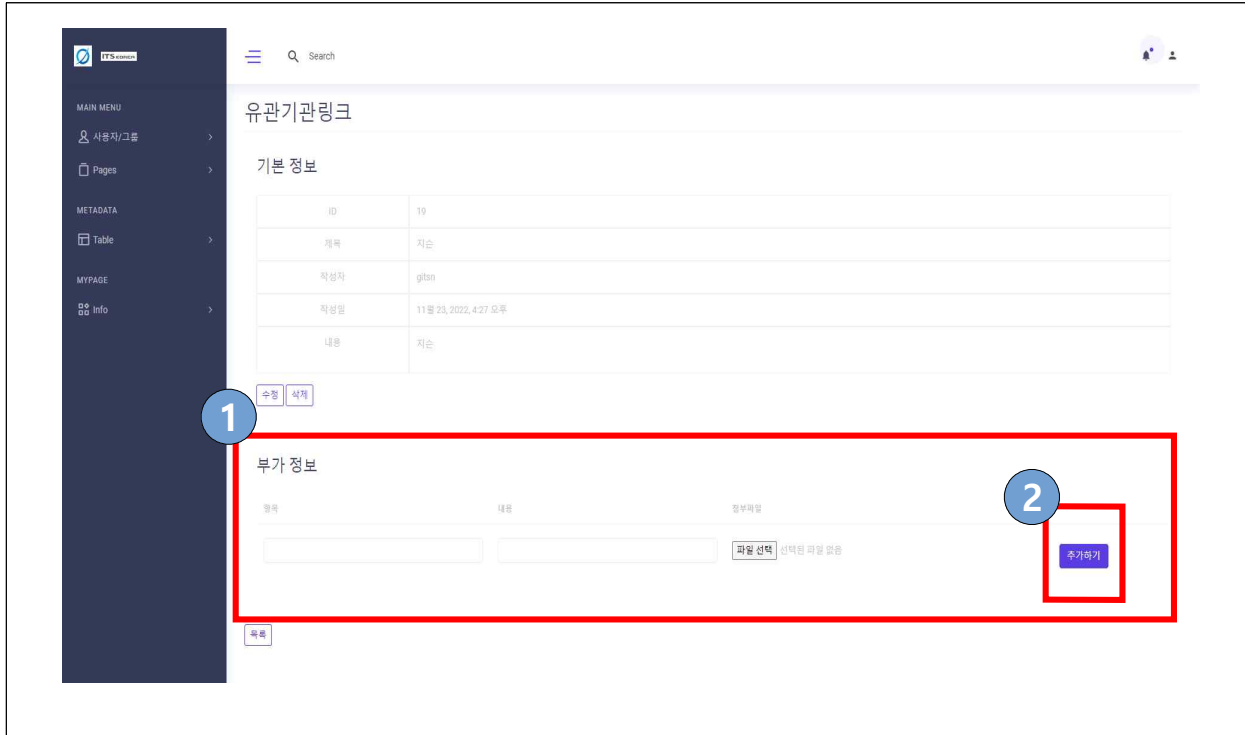
11.5.38. 유관기관 링크 정보 수정



화면 개요	유관기관 링크 정보 수정 화면	
화면 설명	1	등록되어있는 유관기관 링크 기본 정보
	2	‘글 생성 및 부가정보입력’ 버튼
사용 방법	1. 메뉴 : 유관기관 링크 정보 > ‘수정’ 버튼 2. 초기화면 - 유관기관 링크의 기본 정보 수정 가능 3. 선택화면 - ‘글생성 및 부가정보입력’ 버튼 클릭 후 내용 저장	
특기 사항	편집기의 링크 버튼()을 통해서 보이는 글자와 실제 url 링크를 다르게 입력할 수 있음. 관련사이트 링크와 사용법 같음	

(표.67) 유관기관 링크 정보 수정

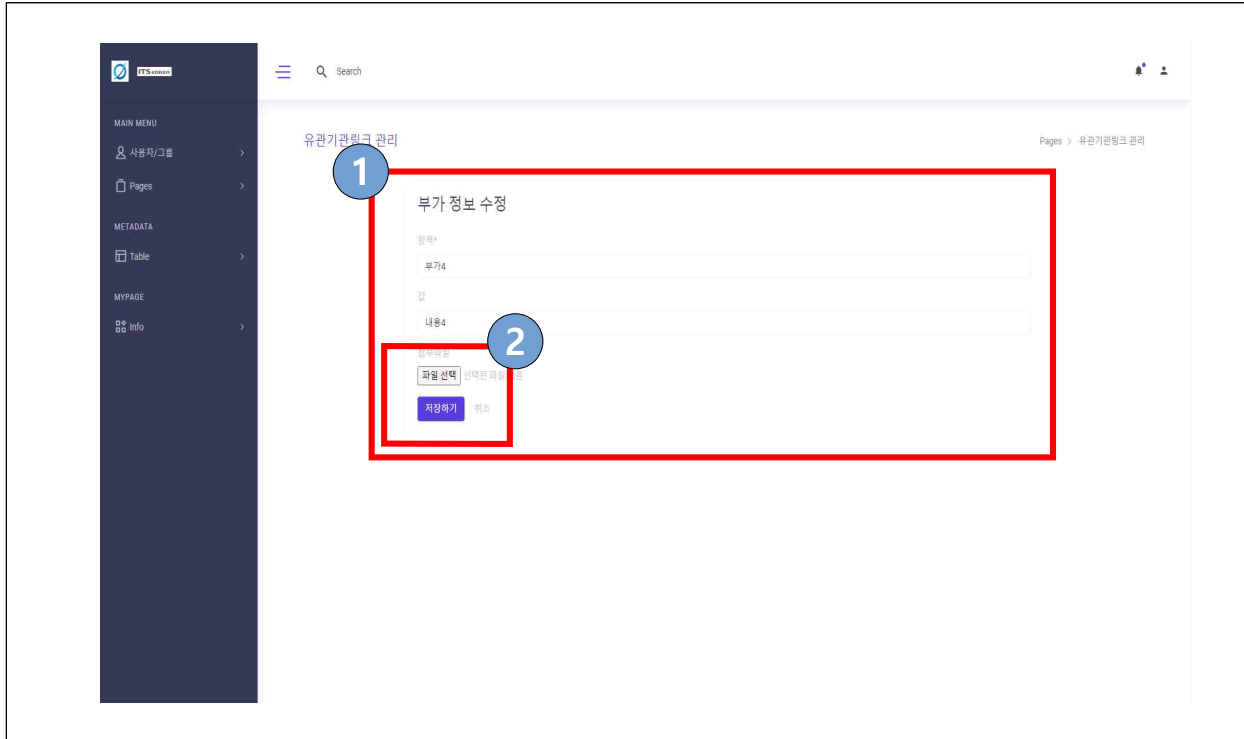
11.5.39. 유관기관 링크 부가정보 추가, 삭제



화면 개요	유관기관 링크 부가정보 추가, 삭제 화면	
화면 설명	1	등록되어있는 유관기관 링크 부가정보
	2	추가, 수정, 삭제 버튼
사용 방법	1. 메뉴 : 유관기관 링크 정보 > 부가정보 '추가' 버튼 2. 초기화면 - 유관기관 링크의 부가정보 추가 및 삭제 가능 3. 선택화면 - 추가하기 버튼 클릭 후 내용 저장 가능	
특기 사항	관련사이트 링크와 사용법 같음	

(표.68) 유관기관 링크 부가정보 추가,삭제

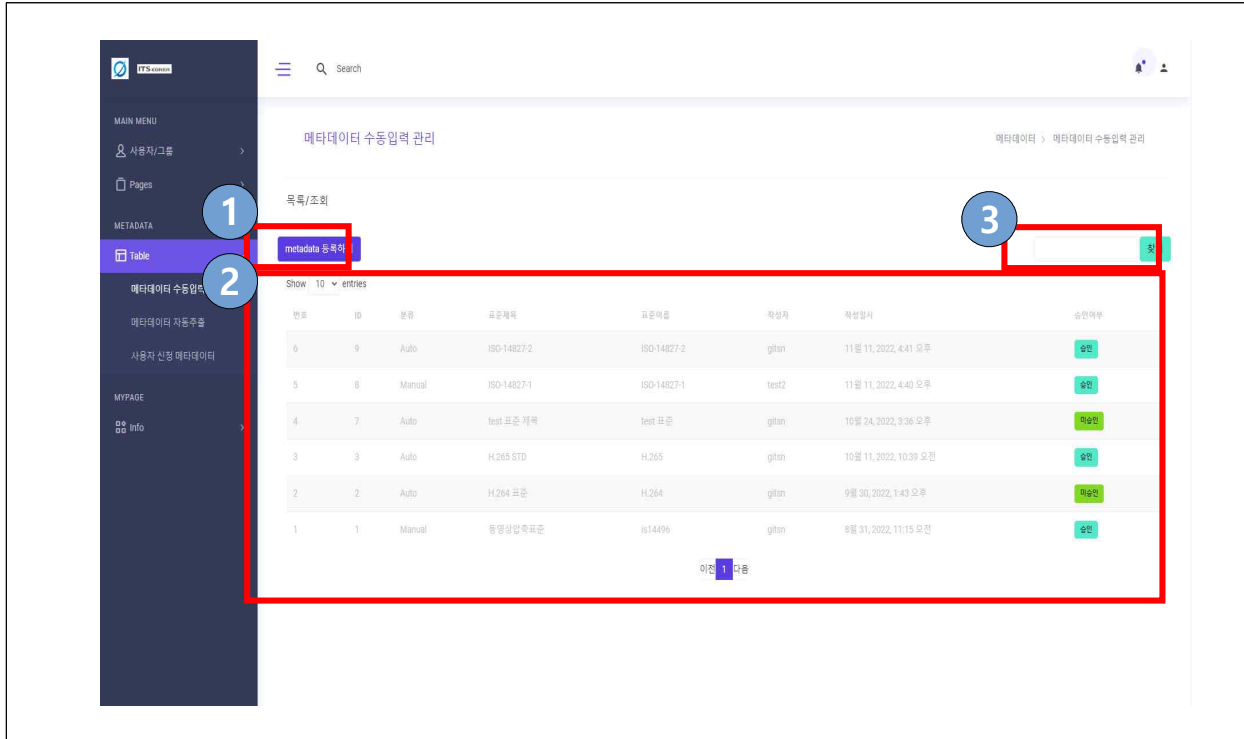
11.5.40. 유관기관 링크 부가정보 수정



화면 개요	유관기관 링크 부가정보 수정 화면	
화면 설명	1	등록되어있는 유관기관 링크 부가정보 수정
	2	파일 첨부 버튼 및 저장 버튼
사용 방법	1. 메뉴 : 유관기관 링크 부가정보 > 부가정보 '수정' 버튼 2. 초기화면 - 유관기관 링크의 부가정보 수정 가능 - 파일 첨부 기능, 파일 삭제 기능 3. 선택화면 - 저장하기 버튼 클릭 후 수정 사항 저장 가능	
특기 사항	파일이 첨부되어 있는 경우 '취소' 를 선택하면 첨부된 파일을 삭제 가능 관련사이트 링크와 사용법 같음	

(표.69) 유관기관 링크 부가정보 수정

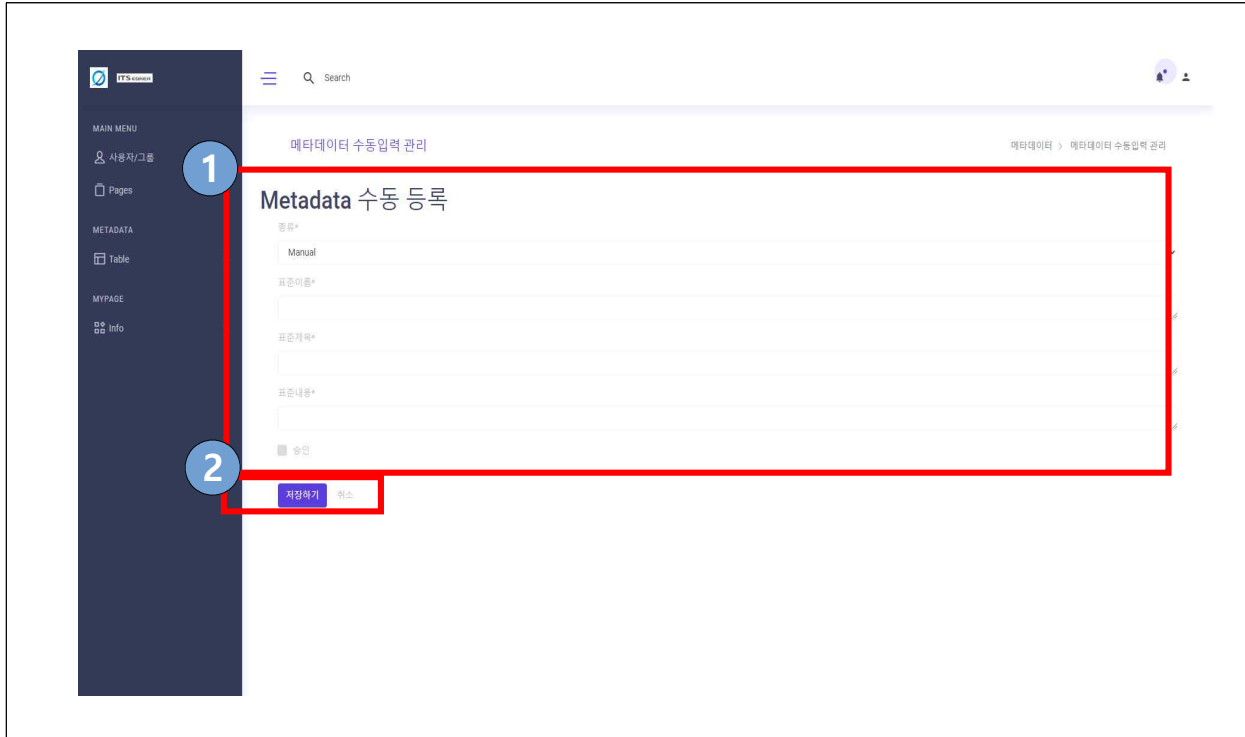
11.5.41. 메타데이터 수동입력 관리



화면 개요	메타데이터 수동입력 관리 화면	
화면 설명	1	메타데이터 수동입력 등록 버튼
	2	전체 메타데이터 수동입력 목록 조회
	3	키워드로 검색 기능
사용 방법	1. 메뉴 : METADATA > Table > 메타데이터 수동입력	
	2. 초기화면 - 메타데이터 수동입력 목록 조회 - 다양한 키워드(표준번호, 표준제목, 표준내용, 작성자)로 편리하게 메타데이터 검색 가능	
	3. 선택화면 - 표준제목 선택 시 메타데이터 수동입력 정보 조회 가능	
특기 사항	없음	

(표.70) 메타데이터 수동입력 관리

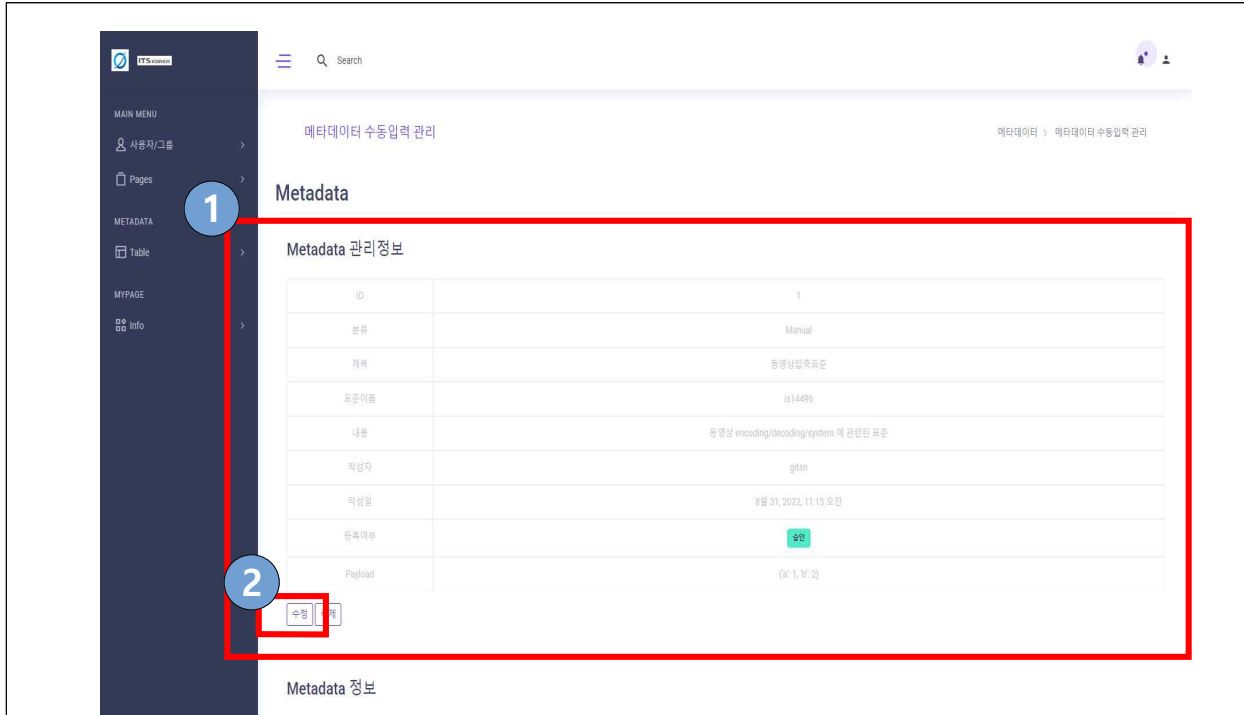
11.5.42. 메타데이터 수동입력 등록



화면 개요	메타데이터 수동입력 등록 화면	
화면 설명	1	메타데이터 수동입력 표준이름, 제목, 내용 입력
	1	승인 체크 표시
	2	'저장하기' 버튼
사용 방법	1. 메뉴 : 메타데이터 수동입력 목록 > 'metadata 등록하기' 버튼	
	2. 초기화면 - 메타데이터 수동입력 표준제목, 표준번호, 표준내용 입력 - 등록 후 사용자 화면에 바로 표시를 원할 때 승인 체크 표시	
특기 사항	3. 선택화면 - 저장하기 버튼 클릭 후 생성	
	메타데이터 수동 등록에서는 메타데이터 관리정보를 먼저 입력하고, 저장하기 버튼 선택 이후 실제 메타데이터 정보를 입력함	

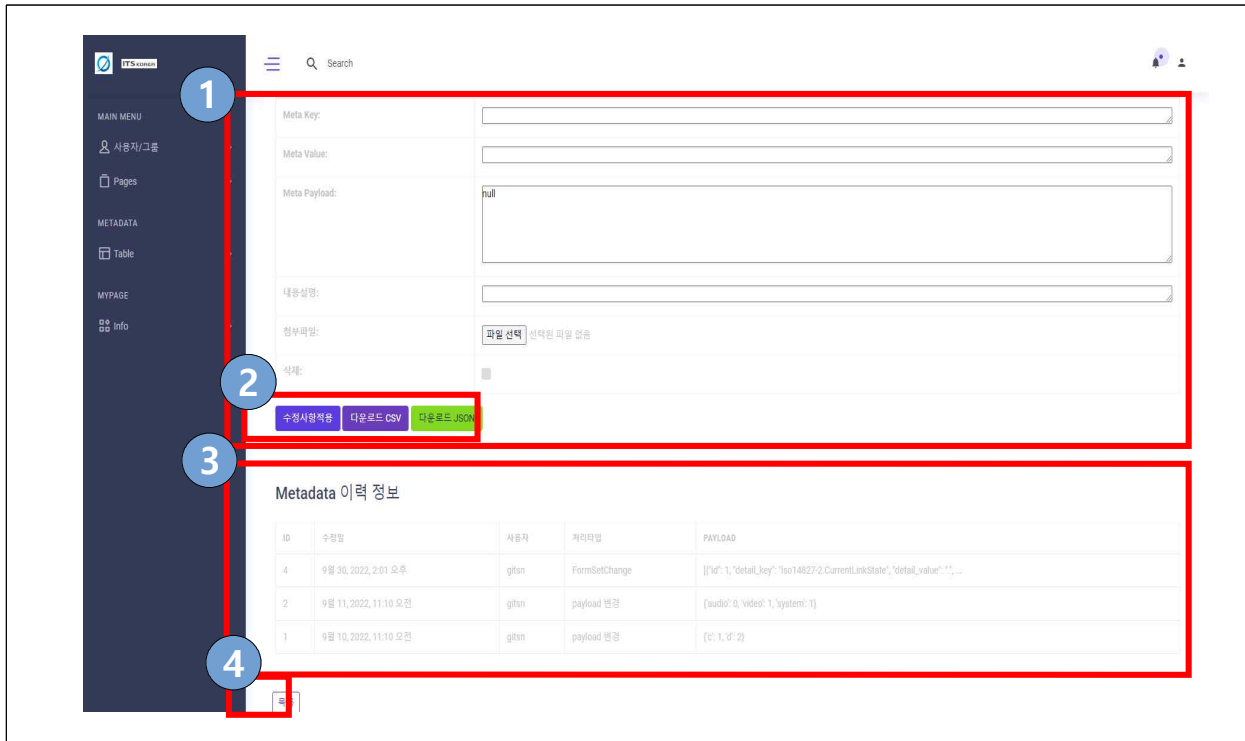
(표.71) 메타데이터 수동입력 등록

11.5.43. 메타데이터 수동입력 정보 조회



화면 개요	메타데이터 수동입력 정보 조회 화면 #1	
화면 설명	1	메타데이터 수동입력의 관리 정보
	2	수정, 삭제 버튼
사용 방법	1. 메뉴 : 메타데이터 수동입력 목록 조회 > 메타데이터 수동입력 정보 2. 초기화면 - 메타데이터 수동입력의 관리 정보 확인 가능 - 수정, 삭제 버튼을 통해 메타데이터 관리정보를 수정하고 삭제 가능	
특기 사항	없음	

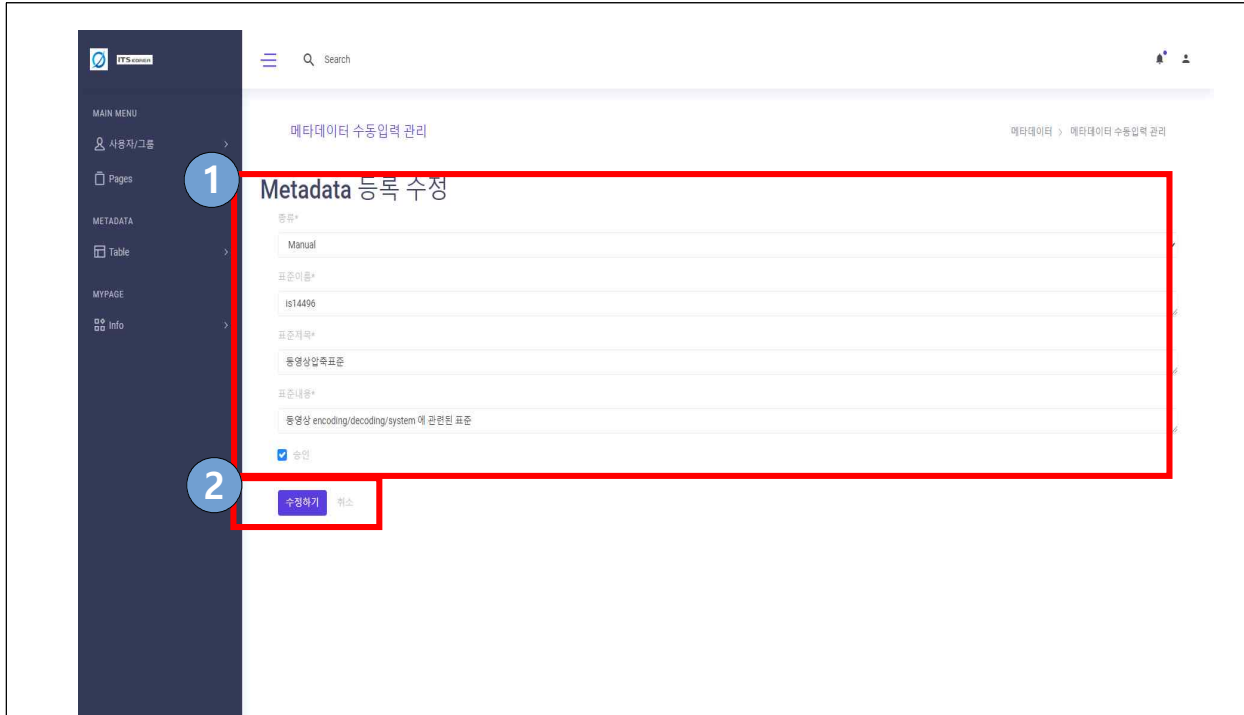
(표.72) 메타데이터 수동입력 정보 조회 #1



화면 개요	메타데이터 수동입력 정보 조회 화면 #2	
화면 설명	1	메타데이터 수동입력의 정보
	2	수정사항 적용, 다운로드 CSV, 다운로드 JSON 버튼
	3	메타데이터 이력 정보
	4	'목록' 버튼
사용 방법	<ol style="list-style-type: none"> 메뉴 : 메타데이터 수동입력 목록 조회 > 메타데이터 수동입력 정보 초기화면 <ul style="list-style-type: none"> 메타데이터 수동입력 관리정보 확인 가능 메타데이터 정보 확인 가능 Meta Key, Meta value Meta Payload, 내용설명, 첨부파일 등 수동으로 메타데이터 정보를 입력한 후 수정사항 적용 버튼 선택 시 메타데이터 저장 가능 다운로드 CSV, JSON 버튼 선택 시 파일로 다운로드 가능 메타데이터의 이력 정보 확인 가능 목록 버튼 클릭 시 목록 조회 페이지로 이동 	
특기 사항	없음	

(표.73) 메타데이터 수동입력 정보 조회 #2

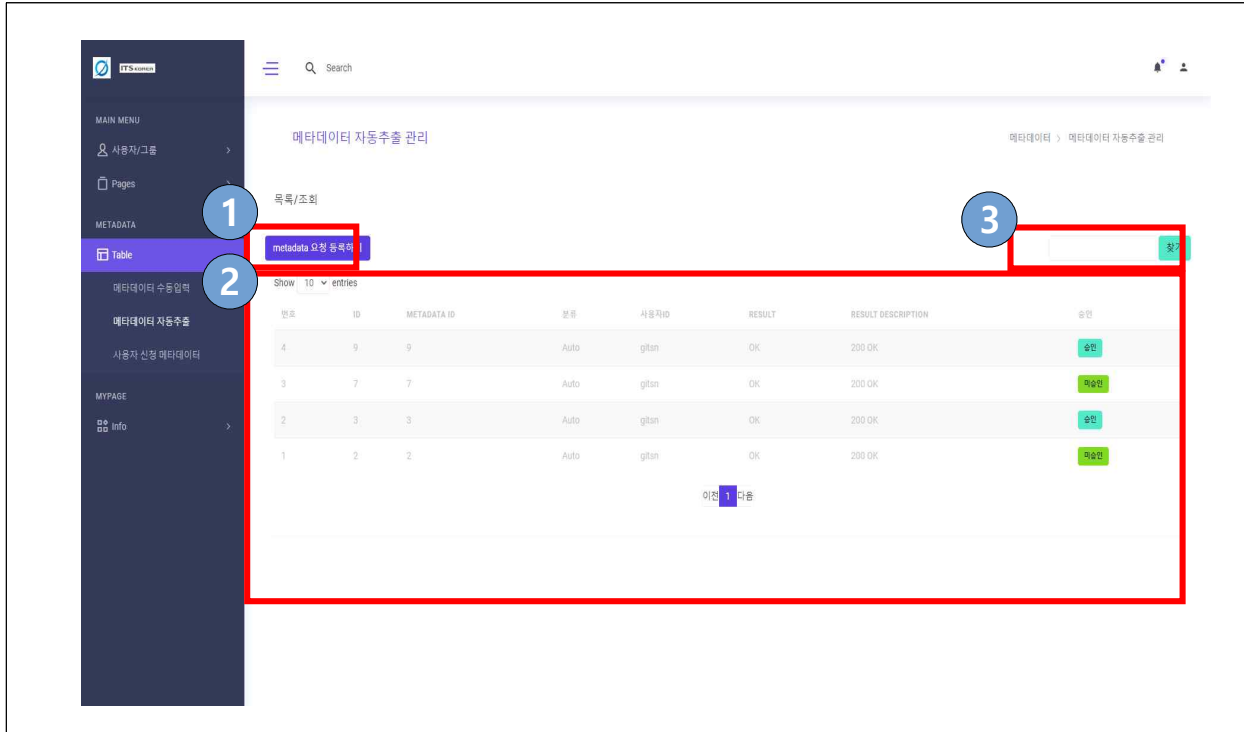
11.5.44. 메타데이터 수동입력 정보 수정



화면 개요	메타데이터 수동입력 정보 수정 화면	
화면 설명	1	등록되어있는 메타데이터 수동입력 기본 정보
	2	'수정하기' 버튼
사용 방법	1. 메뉴 : 메타데이터 수동입력 정보 > '수정' 버튼 2. 초기화면 - 메타데이터 수동입력의 관리정보 수정 가능 - 승인/미승인 체크박스로 사용자 화면에 표시/미표시 선택 가능 3. 선택화면 - 수정하기 버튼 클릭 후 저장	
특기 사항	없음	

(표.74) 메타데이터 수동입력 정보 수정

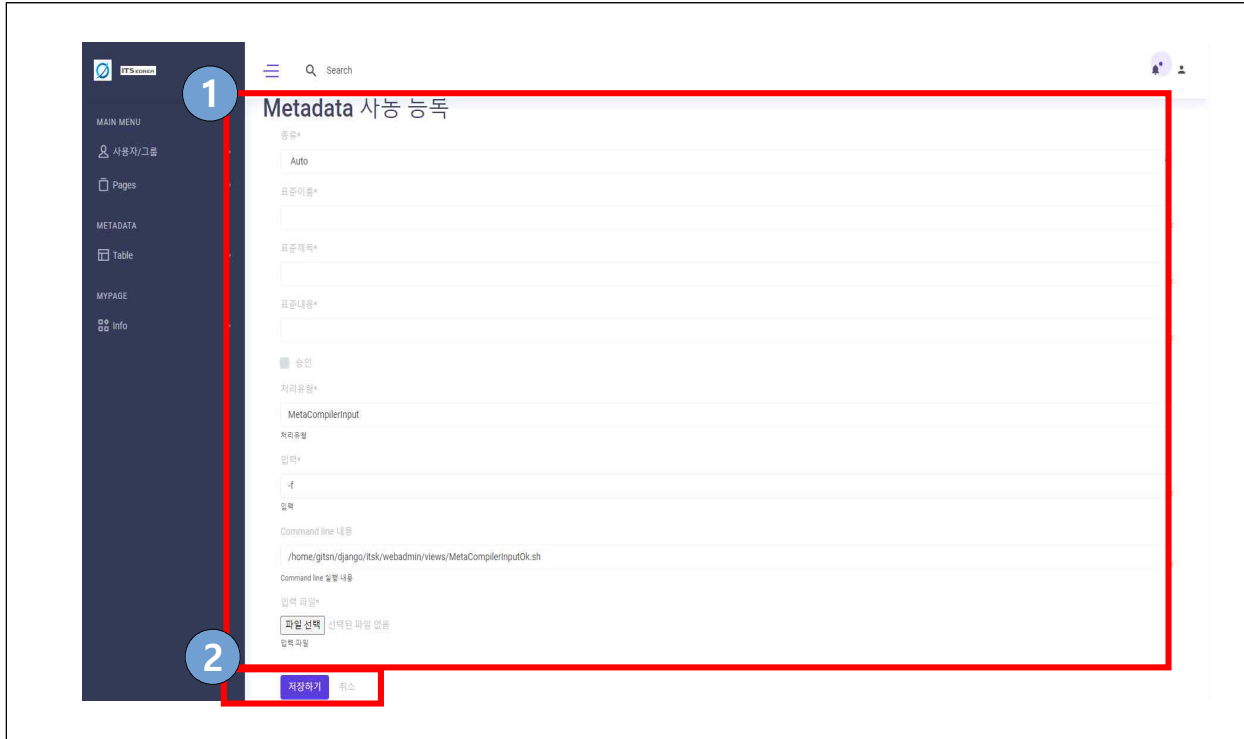
11.5.45. 메타데이터 자동추출 관리



화면 개요	메타데이터 자동추출 관리 화면	
화면 설명	1	메타데이터 자동추출 등록 버튼
	2	전체 메타데이터 자동추출 목록 조회
	3	키워드로 검색 기능
사용 방법	1. 메뉴 : METADATA > Table > 메타데이터 자동추출	
	2. 초기화면 - 메타데이터 자동추출 목록 조회 - 키워드(Result, Result Description)로 편리하게 자동추출 목록 검색 가능	
특기 사항	3. 선택화면 - 번호 선택 시 해당 메타데이터 자동추출 정보 조회 가능	
	메타데이터 자동추출은 메타데이터 관리시스템을 통해 전용 컴파일러를 구동하고 그 결과로 나온 출력을 메타데이터로 등록하는 방식임. 따라서 메타데이터 자동추출 관리 화면은 메타데이터 자체를 표시하는 화면이 아닌 전용 컴파일러의 구동 결과에 대한 목록을 표시하는 화면임	

(표.75) 메타데이터 자동추출 관리

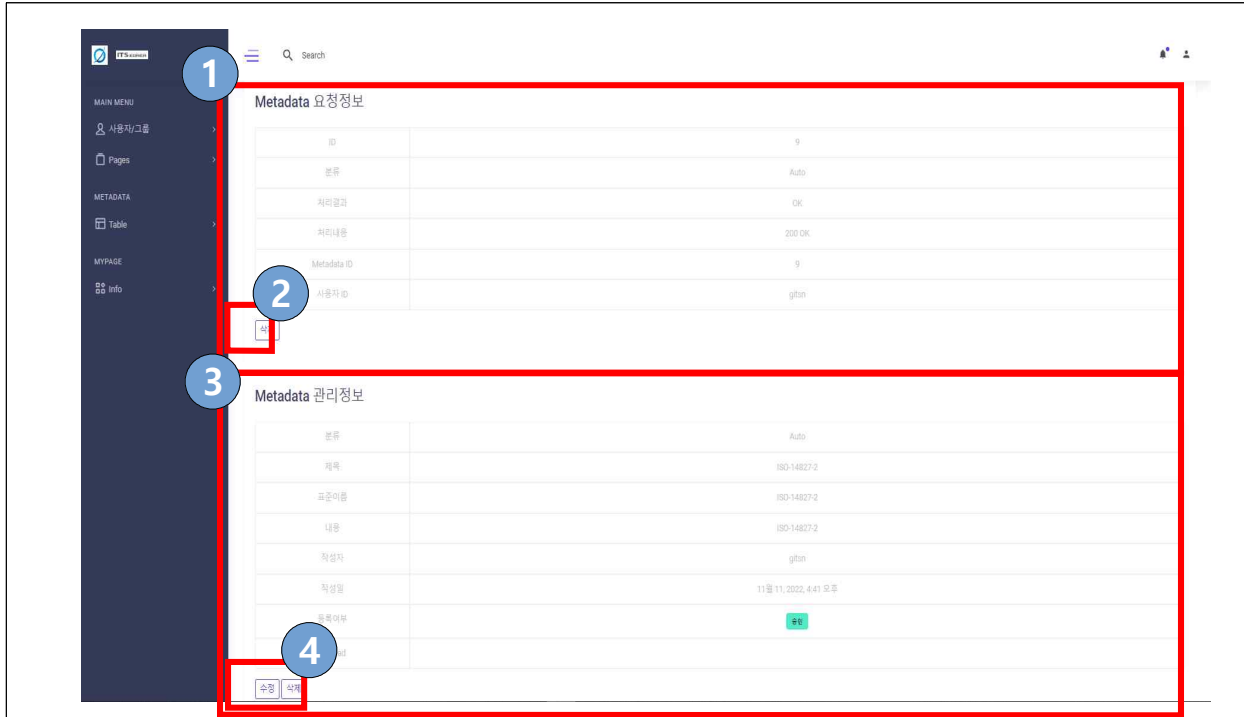
11.5.46. 메타데이터 자동추출 등록



화면 개요	메타데이터 자동추출 등록 화면	
화면 설명	1	메타데이터 자동추출 표준제목, 표준번호, 표준내용 입력
	1	승인 체크 표시
	1	처리유형, 입력, Command line 내용 입력
	2	'저장하기' 버튼
사용 방법	1. 메뉴 : 메타데이터 자동추출 목록 > 'metadata 요청 등록하기' 버튼 2. 초기화면 - 메타데이터 자동추출 표준제목, 표준번호, 표준내용 입력 - 등록 후 사용자 화면에 바로 표시를 원할 때 승인 선택 - 처리유형, Command line 내용은 초기값(default) 그대로 사용 - 입력 파일에 메타데이터를 추출할 ASN.1 파일을 선택	
특기 사항	없음	

(표.76) 메타데이터 자동추출 등록

11.5.47. 메타데이터 자동추출 정보 조회

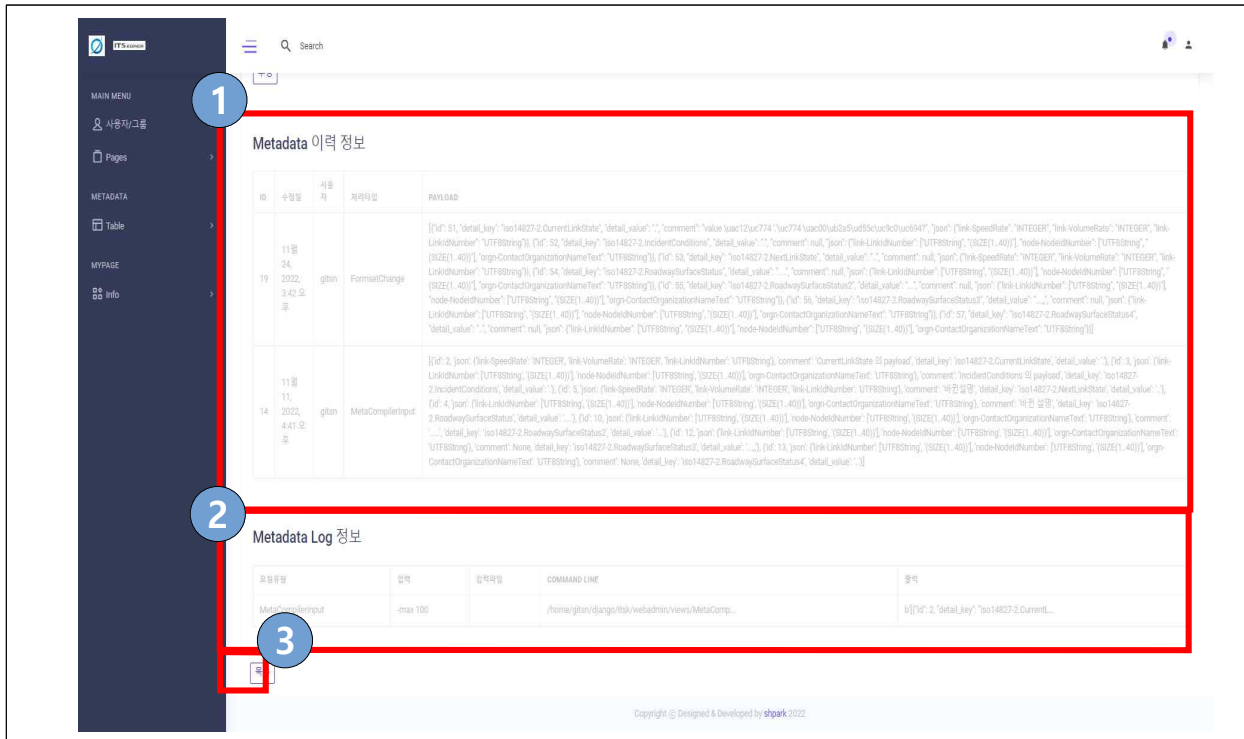


화면 개요	메타데이터 자동추출 정보 조회 화면 #1	
화면 설명	1	메타데이터 자동추출의 요청 정보
	2	삭제 버튼
	3	메타데이터 자동추출의 관리 정보
	4	수정, 삭제 버튼
사용 방법	1. 메뉴 : 메타데이터 자동추출 목록 조회 > 메타데이터 자동추출 정보	
	2. 초기화면 - 메타데이터 자동추출의 요청 정보 확인 가능 - 메타데이터 요청 정보의 삭제 버튼을 통해 삭제 가능 - 메타데이터 자동추출의 관리 정보 확인 가능 - 메타데이터 관리 정보 수정, 삭제 버튼을 통해 수정 및 삭제 가능	
특기 사항	메타데이터 처리 결과 및 처리 내용에 대한 표 참조	

(표.77) 메타데이터 자동추출 정보 조회 #1

	<p>Metadatas 정보</p> <table border="1"> <thead> <tr> <th>ID</th> <th>KEY</th> <th>VALUE</th> <th>PAYLOAD</th> <th>COMMENT</th> </tr> </thead> <tbody> <tr> <td>51</td> <td>iso14827-2.CurrentLinkState</td> <td>-</td> <td>{link-SpeedRate : INTEGER, link-VolumeRate : INTEGER, link-LinkNumber : UTF8String}</td> <td>value 값이 이 가능한지요?</td> </tr> <tr> <td>52</td> <td>iso14827-2.IncidentConditions</td> <td>-</td> <td>{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}</td> <td></td> </tr> <tr> <td>53</td> <td>iso14827-2.NextLinkState</td> <td>-</td> <td>{link-SpeedRate : INTEGER, link-VolumeRate : INTEGER, link-LinkNumber : UTF8String}</td> <td></td> </tr> <tr> <td>54</td> <td>iso14827-2.RoadwaySurfaceStatus</td> <td>-</td> <td>{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}</td> <td></td> </tr> <tr> <td>55</td> <td>iso14827-2.RoadwaySurfaceStatus2</td> <td>-</td> <td>{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}</td> <td></td> </tr> <tr> <td>56</td> <td>iso14827-2.RoadwaySurfaceStatus3</td> <td>-</td> <td>{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}</td> <td></td> </tr> <tr> <td>57</td> <td>iso14827-2.RoadwaySurfaceStatus4</td> <td>-</td> <td>{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}</td> <td></td> </tr> </tbody> </table> <p>Metadatas 이력 정보</p> <table border="1"> <thead> <tr> <th>ID</th> <th>수정일</th> <th>수정자</th> <th>처리내역</th> <th>PAYLOAD</th> </tr> </thead> <tbody> <tr> <td>11월 24, 2022, 2:42 오후</td> <td>glsn</td> <td>FormatChange</td> <td></td> <td>[[{"id": 51, "detail_key": "iso14827-2.CurrentLinkState", "detail_value": "", "comment": "value issue", "payload": "{link-SpeedRate : INTEGER, link-VolumeRate : INTEGER, link-LinkNumber : UTF8String}", "payload_size": 40}, {"id": 52, "detail_key": "iso14827-2.IncidentConditions", "detail_value": "", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String", "payload_size": 53}, {"id": 53, "detail_key": "iso14827-2.NextLinkState", "detail_value": "", "comment": "null", "payload": "{link-SpeedRate : INTEGER, link-VolumeRate : INTEGER, link-LinkNumber : UTF8String}", "payload_size": 54}, {"id": 54, "detail_key": "iso14827-2.RoadwaySurfaceStatus", "detail_value": "...", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String", "payload_size": 55}, {"id": 55, "detail_key": "iso14827-2.RoadwaySurfaceStatus2", "detail_value": "...", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String", "payload_size": 56}, {"id": 56, "detail_key": "iso14827-2.RoadwaySurfaceStatus3", "detail_value": "...", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String", "payload_size": 57}, {"id": 57, "detail_key": "iso14827-2.RoadwaySurfaceStatus4", "detail_value": "...", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String"}]]</td> </tr> </tbody> </table>	ID	KEY	VALUE	PAYLOAD	COMMENT	51	iso14827-2.CurrentLinkState	-	{link-SpeedRate : INTEGER, link-VolumeRate : INTEGER, link-LinkNumber : UTF8String}	value 값이 이 가능한지요?	52	iso14827-2.IncidentConditions	-	{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}		53	iso14827-2.NextLinkState	-	{link-SpeedRate : INTEGER, link-VolumeRate : INTEGER, link-LinkNumber : UTF8String}		54	iso14827-2.RoadwaySurfaceStatus	-	{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}		55	iso14827-2.RoadwaySurfaceStatus2	-	{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}		56	iso14827-2.RoadwaySurfaceStatus3	-	{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}		57	iso14827-2.RoadwaySurfaceStatus4	-	{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}		ID	수정일	수정자	처리내역	PAYLOAD	11월 24, 2022, 2:42 오후	glsn	FormatChange		[[{"id": 51, "detail_key": "iso14827-2.CurrentLinkState", "detail_value": "", "comment": "value issue", "payload": "{link-SpeedRate : INTEGER, link-VolumeRate : INTEGER, link-LinkNumber : UTF8String}", "payload_size": 40}, {"id": 52, "detail_key": "iso14827-2.IncidentConditions", "detail_value": "", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String", "payload_size": 53}, {"id": 53, "detail_key": "iso14827-2.NextLinkState", "detail_value": "", "comment": "null", "payload": "{link-SpeedRate : INTEGER, link-VolumeRate : INTEGER, link-LinkNumber : UTF8String}", "payload_size": 54}, {"id": 54, "detail_key": "iso14827-2.RoadwaySurfaceStatus", "detail_value": "...", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String", "payload_size": 55}, {"id": 55, "detail_key": "iso14827-2.RoadwaySurfaceStatus2", "detail_value": "...", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String", "payload_size": 56}, {"id": 56, "detail_key": "iso14827-2.RoadwaySurfaceStatus3", "detail_value": "...", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String", "payload_size": 57}, {"id": 57, "detail_key": "iso14827-2.RoadwaySurfaceStatus4", "detail_value": "...", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String"}]]
ID	KEY	VALUE	PAYLOAD	COMMENT																																															
51	iso14827-2.CurrentLinkState	-	{link-SpeedRate : INTEGER, link-VolumeRate : INTEGER, link-LinkNumber : UTF8String}	value 값이 이 가능한지요?																																															
52	iso14827-2.IncidentConditions	-	{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}																																																
53	iso14827-2.NextLinkState	-	{link-SpeedRate : INTEGER, link-VolumeRate : INTEGER, link-LinkNumber : UTF8String}																																																
54	iso14827-2.RoadwaySurfaceStatus	-	{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}																																																
55	iso14827-2.RoadwaySurfaceStatus2	-	{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}																																																
56	iso14827-2.RoadwaySurfaceStatus3	-	{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}																																																
57	iso14827-2.RoadwaySurfaceStatus4	-	{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}																																																
ID	수정일	수정자	처리내역	PAYLOAD																																															
11월 24, 2022, 2:42 오후	glsn	FormatChange		[[{"id": 51, "detail_key": "iso14827-2.CurrentLinkState", "detail_value": "", "comment": "value issue", "payload": "{link-SpeedRate : INTEGER, link-VolumeRate : INTEGER, link-LinkNumber : UTF8String}", "payload_size": 40}, {"id": 52, "detail_key": "iso14827-2.IncidentConditions", "detail_value": "", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String", "payload_size": 53}, {"id": 53, "detail_key": "iso14827-2.NextLinkState", "detail_value": "", "comment": "null", "payload": "{link-SpeedRate : INTEGER, link-VolumeRate : INTEGER, link-LinkNumber : UTF8String}", "payload_size": 54}, {"id": 54, "detail_key": "iso14827-2.RoadwaySurfaceStatus", "detail_value": "...", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String", "payload_size": 55}, {"id": 55, "detail_key": "iso14827-2.RoadwaySurfaceStatus2", "detail_value": "...", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String", "payload_size": 56}, {"id": 56, "detail_key": "iso14827-2.RoadwaySurfaceStatus3", "detail_value": "...", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String", "payload_size": 57}, {"id": 57, "detail_key": "iso14827-2.RoadwaySurfaceStatus4", "detail_value": "...", "comment": "null", "payload": "{link-LinkNumber : [UTF8String, (SIZE)1..40]}, node-NodeNumber : [UTF8String, (SIZE)1..40]}, org-ContactOrganizationNameText : UTF8String"}]]																																															
<p>화면 개요</p>	<p>메타데이터 자동추출 정보 조회 화면 #2</p>																																																		
<p>화면 설명</p>	<table border="1"> <tr> <td>1</td> <td>메타데이터 자동추출의 정보</td> </tr> <tr> <td>2</td> <td>수정 버튼</td> </tr> <tr> <td>3</td> <td>메타데이터 이력 정보</td> </tr> <tr> <td></td> <td></td> </tr> <tr> <td></td> <td></td> </tr> </table>	1	메타데이터 자동추출의 정보	2	수정 버튼	3	메타데이터 이력 정보																																												
1	메타데이터 자동추출의 정보																																																		
2	수정 버튼																																																		
3	메타데이터 이력 정보																																																		
<p>사용 방법</p>	<ol style="list-style-type: none"> 메뉴 : 메타데이터 자동추출 목록 조회 > 메타데이터 자동추출 정보 초기화면 <ul style="list-style-type: none"> 메타데이터 자동추출 처리 정보 확인 가능 메타데이터 정보 수동으로 수정 가능 메타데이터의 이력 정보 확인 가능 선택화면 <ul style="list-style-type: none"> '수정' 버튼을 클릭하여 메타데이터 수정 가능 																																																		
<p>특기 사항</p>	<p>없음</p>																																																		

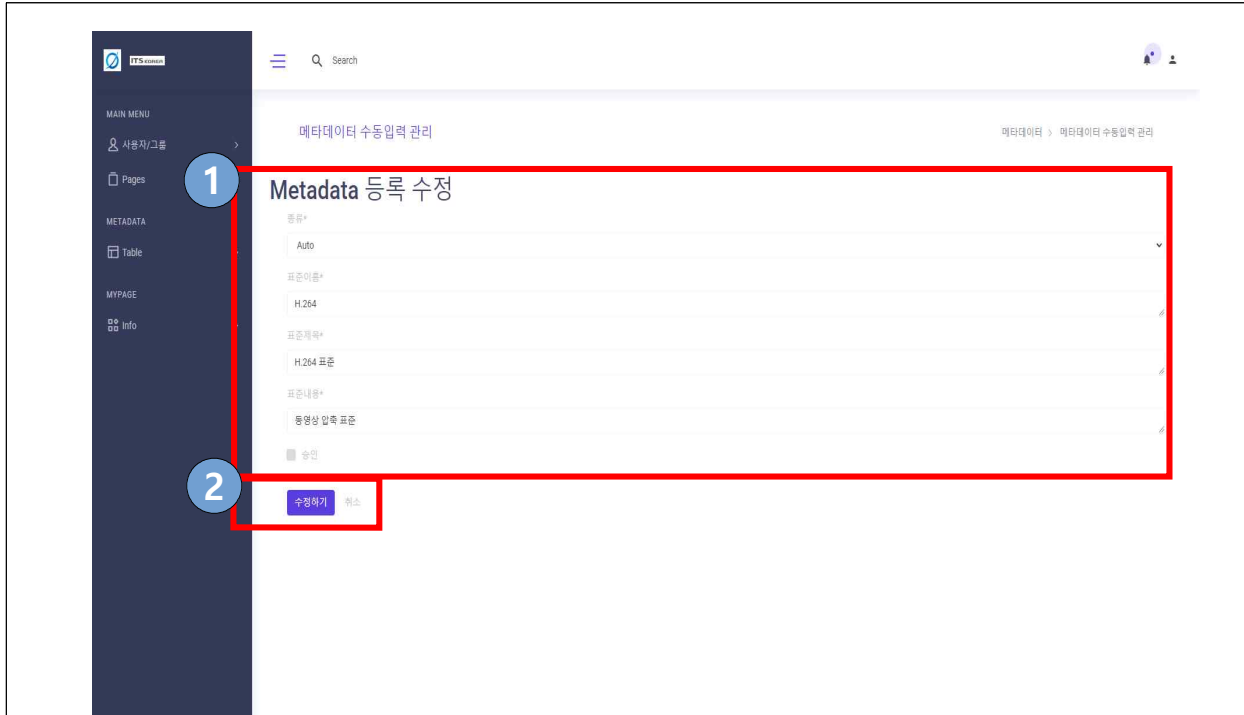
(표.78) 메타데이터 자동추출 정보 조회 #2



화면 개요	메타데이터 자동추출 정보 조회 화면 #3	
화면 설명	1	메타데이터 자동추출의 정보
	2	메타데이터의 Log 정보
	3	'목록' 버튼
사용 방법	1. 메뉴 : 메타데이터 자동추출 목록 조회 > 메타데이터 자동추출 정보 2. 초기화면 - 메타데이터 자동추출의 이력 정보 확인 가능 - 메타데이터의 Log 정보 확인 가능 - 목록 버튼 클릭 시 메타데이터 자동추출 목록 조회 페이지로 이동	
특기 사항	없음	

(표.79) 메타데이터 자동추출 정보 조회 #3

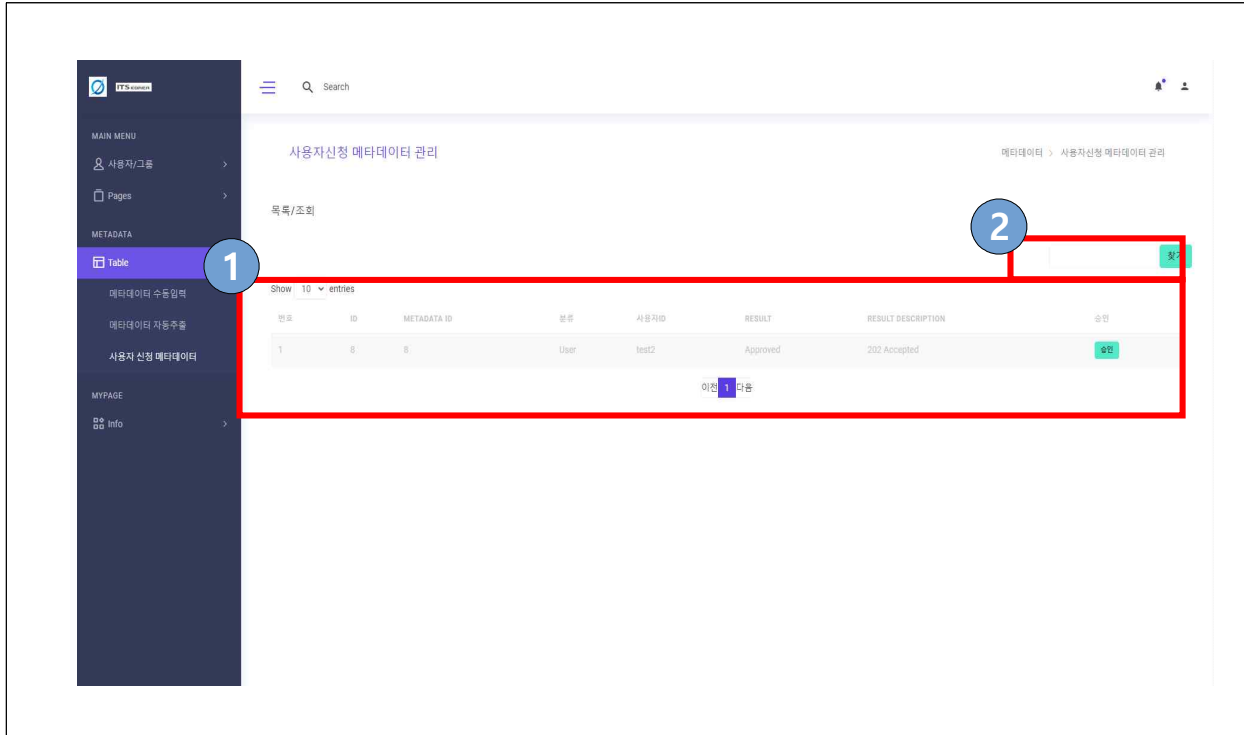
11.5.48. 메타데이터 자동추출 정보 수정



화면 개요	메타데이터 자동추출 정보 수정 화면	
화면 설명	1	등록되어있는 메타데이터 자동추출 기본 정보
	2	‘수정하기’ 버튼
사용 방법	1. 메뉴 : 메타데이터 자동추출 정보 > ‘수정’ 버튼 2. 초기화면 - 메타데이터 자동추출의 기본 정보 수정 가능 - 승인 선택 시 사용자 화면에 해당 메타데이터를 바로 표시 가능 3. 선택화면 - 수정하기 버튼 클릭 후 저장	
특기 사항	정보 수정은 메타데이터 수동입력 수정과 동일하게 메타데이터 관리정보 및 메타데이터 정보 수정을 통해서 가능	

(표.80) 메타데이터 자동추출 정보 수정

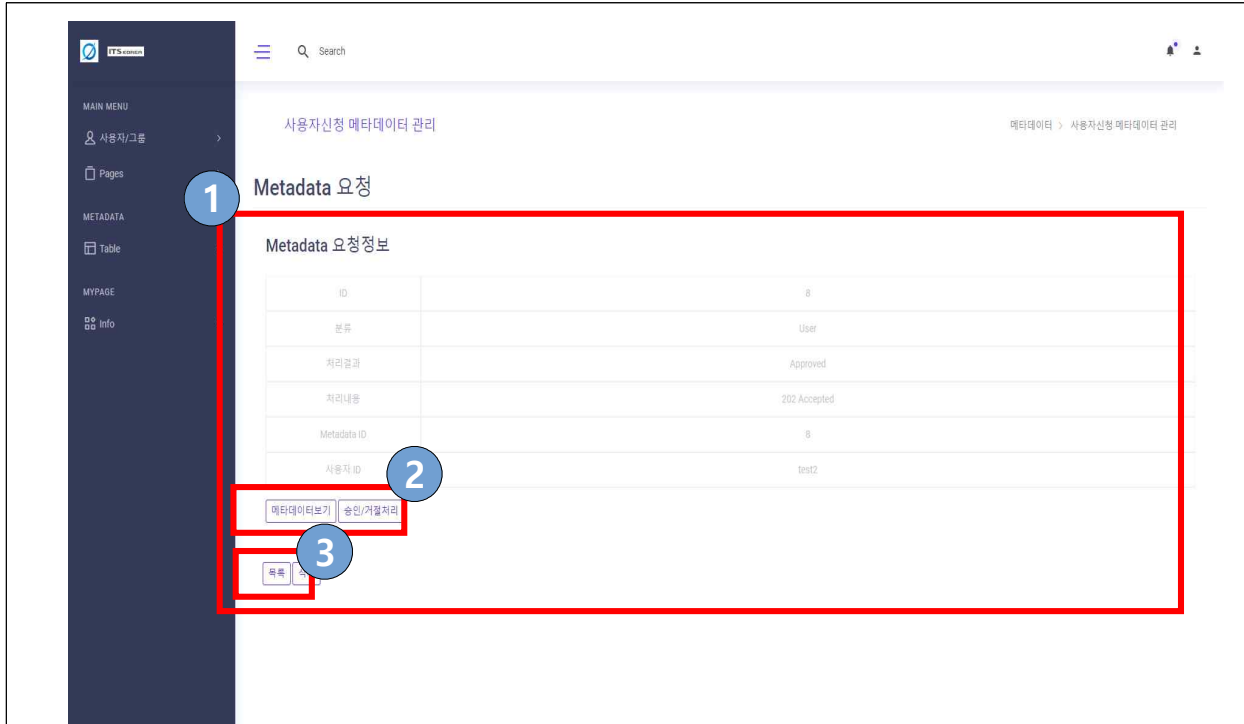
11.5.49. 사용자신청 메타데이터 관리



화면 개요	사용자신청 메타데이터 관리 화면	
화면 설명	1	전체 사용자신청 메타데이터 목록 조회
	2	키워드로 검색 기능
사용 방법	1. 메뉴 : METADATA > Table > 사용자신청 메타데이터 2. 초기화면 - 사용자신청 메타데이터 목록 조회 - 키워드(Result, Result Description)로 편리하게 사용자신청 목록 검색 가능 3. 선택화면 - 번호 선택 시 사용자신청 메타데이터 정보 조회 가능	
특기 사항	없음	

(표.81) 사용자신청 메타데이터 관리

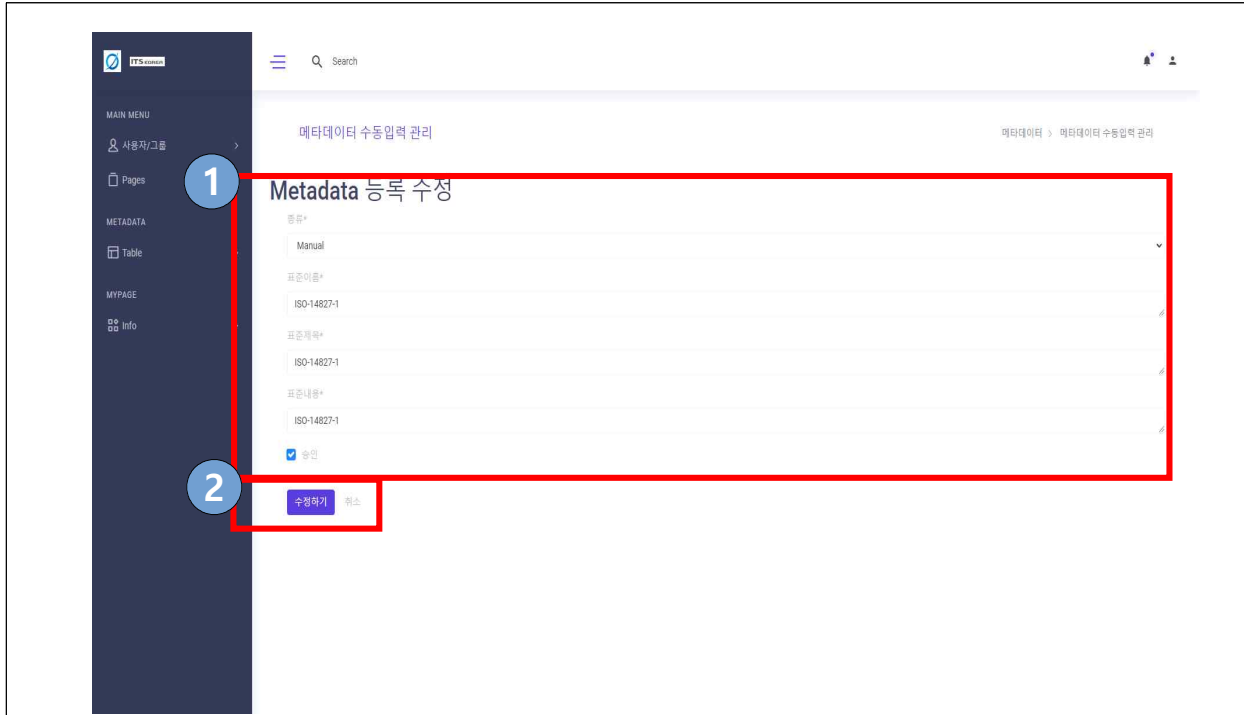
11.5.50. 사용자신청 메타데이터 정보 조회



화면 개요	사용자신청 메타데이터 정보 조회 화면	
화면 설명	1	사용자신청 메타데이터의 요청 정보
	2	메타데이터보기, 승인/거절처리 버튼
	3	목록, 삭제 버튼
사용 방법	1. 메뉴 : 사용자신청 메타데이터 목록 조회 > 사용자신청 메타데이터 정보 2. 초기화면 - 사용자신청 메타데이터 요청 정보 확인 가능 - 메타데이터보기로 사용자가 신청한 메타데이터를 세부적으로 확인 및 수정 가능 - 승인/거절처리 버튼을 통해 사용자의 메타데이터 신청을 승인 및 거절 처리 가능 - 목록 버튼 클릭 시 사용자신청 메타데이터 목록 조회 페이지로 이동	
특기 사항	없음	

(표.82) 사용자신청 메타데이터 정보 조회

11.5.51. 사용자신청 메타데이터 정보 수정



화면 개요	사용자신청 메타데이터 정보 수정 화면	
화면 설명	1	등록되어있는 사용자신청 메타데이터 관리 정보
	2	‘수정하기’ 버튼
사용 방법	1. 메뉴 : 사용자신청 메타데이터 정보 > ‘수정’ 버튼 2. 초기화면 - 사용자신청 메타데이터의 관리 정보 수정 가능 3. 선택화면 - 수정하기 버튼 클릭 후 저장	
특기 사항	정보 수정은 메타데이터 수동입력 정보 수정과 동일하게 메타데이터 관리정보 및 메타데이터 정보 수정을 통해서 가능	

(표.83) 사용자신청 메타데이터 정보 수정

11.5.52. 메타데이터 처리 결과 및 내용에 대한 값과 의미

항목	값	의미
처리 결과	'OK'	전용 컴파일러 처리 성공
	'Failed'	전용 컴파일러 처리 실패
	'Pending'	사용자 요청 메타데이터 처리 대기 중
	'Approved'	사용자 요청 메타데이터 승인 성공
	'Rejected'	사용자 요청 메타데이터 승인 거부
처리 내용	'200 OK'	전용 컴파일러 처리 성공
	'201 Created'	사용자 요청 메타데이터 생성
	'202 Accepted'	사용자 요청 메타데이터 승인 성공
	'400 Bad Request'	사용자 요청이 잘못됨
	'406 Not Acceptable'	사용자 요청 메타데이터 승인 거부
	'500 Internal Server Error'	서버 오류로 전용 컴파일러 처리 실패

(표.84) 메타데이터 처리 결과 및 내용