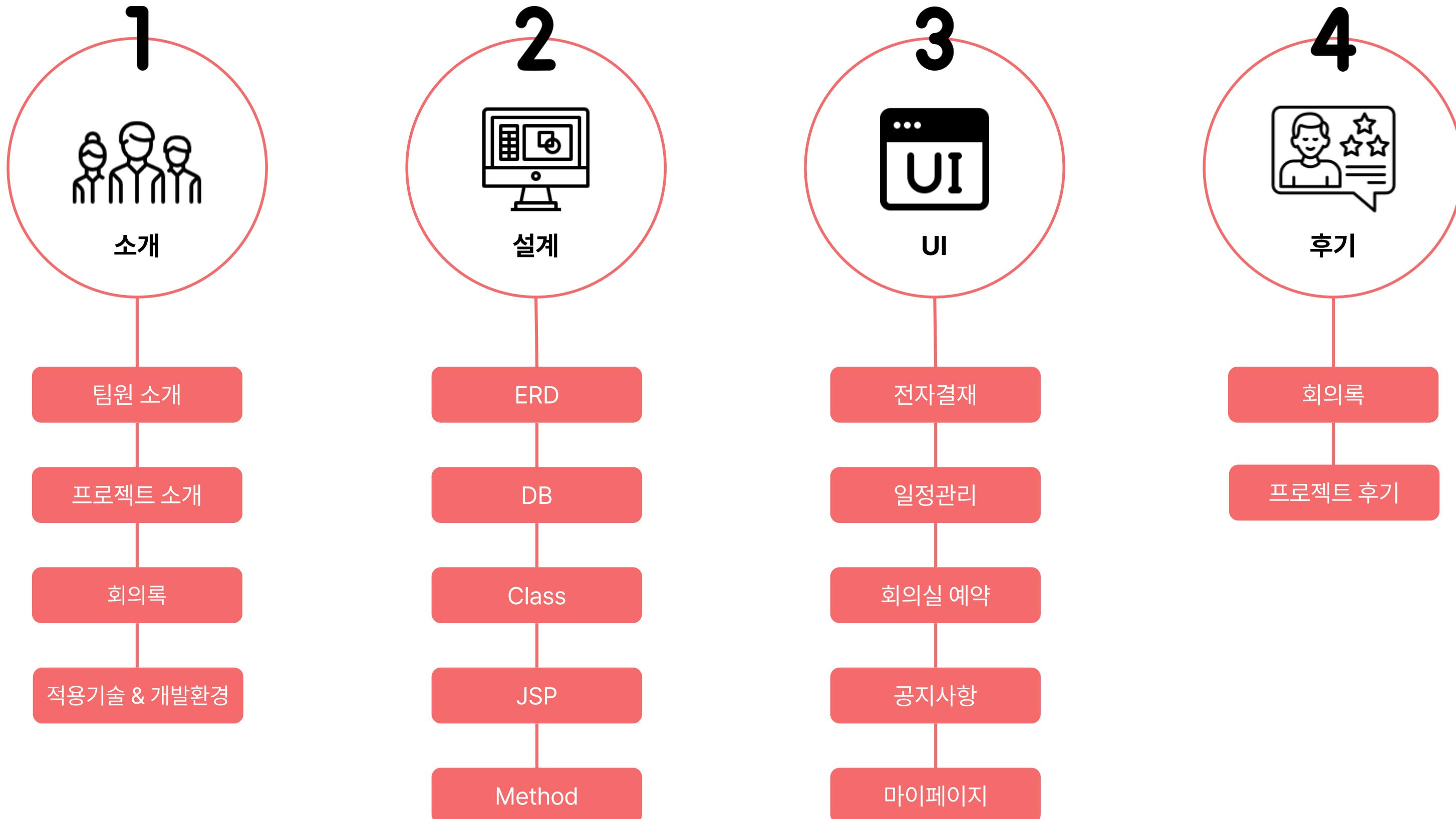




Easy, Minimal, ERP

Dev Pixel







정민규/리더

PPT / 발표



손혜인/리더

CSS



최철민/팀원



임예찬/팀원



김진성/팀원



조수아/팀원



한상경/팀원

| | | | |
|-------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 전자결재 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 일정관리 | | | <input checked="" type="checkbox"/> |
| 회의실 예약 | <input checked="" type="checkbox"/> | | |
| 공지사항 | <input checked="" type="checkbox"/> | | <input checked="" type="checkbox"/> |
| 관리자 & 마이페이지 | <input checked="" type="checkbox"/> | | <input checked="" type="checkbox"/> |
| 로그인 & 회원가입 | | | <input checked="" type="checkbox"/> |
| 소스코드 종합 | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |



프로젝트 소개

Easy, Minimal, ERP

프로젝트 소개

WEB 기반 사내 전자결재 ERP 시스템

프로젝트 방향성

- 사내 전자결재 ERP 시스템 구축을 통한 효율적 서비스 제공
- 개인 일정 관리 및 부서별 업무 기간 공유, 회의실 예약 기능 등을 통한 편리성 제공
- 미니멀 트렌드를 반영한 간단하고 직관적인 UI 제공

2022.11.21

1. 프로젝트 아이디어 도출

- a. 교생 관련 사이트
- b. 영양 & 헬스 관련 사이트
- c. 풋살 팀 매칭 사이트

2. 프로젝트 아이디어 선정

레퍼런스 및 카테고리 부족으로 전자결재 ERP 선정

2022.11.22

1. 자바 클래스 구성 (임예찬, 한상경, 김진성)

1-1 LoginController, BoardController 구성

2. 자바 클래스 구성 간 제작 (임예찬, 한상경)

2-1 LoginController, BoardController 제작

3. DB 설계 (손혜인, 조수아, 정민규, 최철민)

3-1 User, Board 테이블 설계(PK, FK 설정)

4. DB 설계 간 테이블 제작 (손혜인, 조수아)

4-1 User, Board 테이블 제작

5. UI 보수 및 설계(김진성)

6. 카테고리별 업무분담

1. 결재 페이지

- 임예찬, 최철민, 김진성

2. 로그인, 일정관리, 회의실 예약

- 한상경, 정민규

3. 커뮤니티, 마이페이지, 관리자페이지

- 손혜인, 조수아

2022.11.23

1. 카테고리 설정 및 UI 구현(HTML 사용)
2. CSS 적용
3. 리더 설정 (정민규, 손혜인)
4. 역할 분담

2022.11.24

~

DB 설계

2022.11.26

2022.11.27

분담한 역할대로 개인 작업 진행

2022.11.28

SQL, Class 제작 등의 구체적인 업무 분담이 어려워 카테고리별 업무 분담 인원끼리 상의 후 전반적인 기능 구현위주로 재분담

2022.11.29

1. DB설계

필요 테이블 추가(회의실예약관련 테이블, 휴가신청서 관련 테이블)

2. ERD 다이어그램 설계

총 테이블 20개로 설계함

2022.11.30

~

2022.12.05

간단한 진행상황 공유

2022.12.06

1. 각 카테고리 별 진행상황 공유
2. 일정관리 부분은 다른 카테고리 완료 후 진행하기로 함
3. 회의실예약과 공지사항, 카테고리가 일찍 마치면 전자결재로 다같이 진행하기로 결정

2022.12.08

1. 기능이 완료된 소스 합치고 에러상황 공유
2. 일정관리 다시 맡아서 구현하기
3. 회의실예약과 공지사항, 일정관리 카테고리가 일찍 마치면 전자결재로 다같이 진행하기로 결정

2022.12.09

1. 전자결재 제외한 나머지 파트 미비한 부분 마무리

2022.12.10

~

2022.12.13

1. 전자결재 제외한 나머지 파트 소스 종합
2. 미비한 부분 마무리
3. 테스트를 통한 에러 수정

2022.12.14

~

2022.12.16

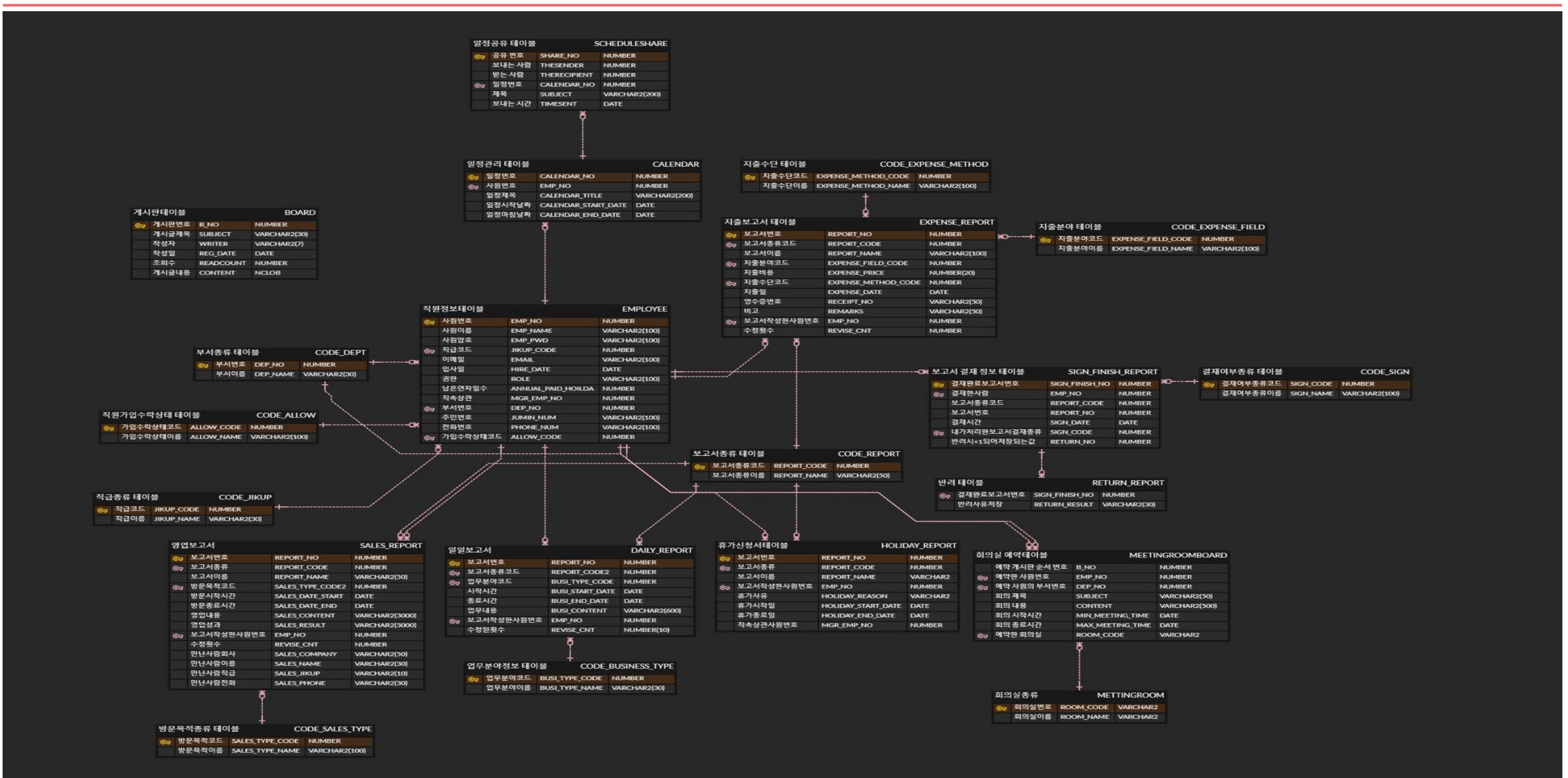
1. 모든 소스 종합 및 테스트
2. 각자 지적사항을 통하여 부족한 부분 수정



2



설계



일일보고서 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|--------------|-----------------|----------|--------------|----|----|------|---------------|---------|--------------------|
| DAILY_REPORT | REPORT_NO | 일일보고서 | 보고서번호 | Y | | | NUMBER | | |
| | REPORT_CODE | | 보고서종류 | | Y | Y | NUMBER | | CODE_REPORT |
| | BUSI_TYPE_CODE | | 업무분야코드 | | Y | Y | NUMBER | | CODE_BUSINESS_TYPE |
| | BUSI_START_DATE | | 시작시간 | | | Y | DATE | | |
| | BUSI_END_DATE | | 종료시간 | | | Y | DATE | | |
| | BUSI_CONTENT | | 업무내용 | | | Y | VARCHAR2(600) | | |
| | EMP_NO | | 보고서 작성한 사원번호 | | Y | Y | NUMBER | | EMPLOYEE |
| | REVISE_CNT | | 수정된 횟수 | | | Y | NUMBER(10) | | |

영업보고서 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|--------------|------------------|----------|------------|----|----|------|----------------|---------|-----------------|
| SALES_REPORT | REPORT_NO | 영업보고서 | 보고서번호 | Y | | | NUMBER | | |
| | REPORT_CODE | | 보고서종류 | | Y | Y | NUMBER | | CODE_REPORT |
| | REPORT_NAME | | 보고서이름 | | | Y | VARCHAR2(100) | | |
| | SALES_TYPE_CODE | | 방문목적코드 | | Y | Y | NUMBER | | CODE_SALES_TYPE |
| | SALES_DATE_START | | 방문시작시간 | | | Y | DATE | | |
| | SALES_DATE_END | | 방문종료시간 | | | Y | DATE | | |
| | SALES_CONTENT | | 영업내용 | | | Y | VARCHAR2(3000) | | |
| | SALES_RESULT | | 영업성과 | | | Y | VARCHAR2(3000) | | |
| | EMP_NO | | 보고서작성한사원번호 | | Y | Y | NUMBER | | EMPLOYEE |
| | REVISE_CNT | | 수정횟수 | | | Y | NUMBER | | |
| | SALES_COMPANY | | 만난사람회사 | | | Y | VARCHAR2(100) | | |
| | SALES_NAME | | 만난사람이름 | | | Y | VARCHAR2(100) | | |
| | SALES_JIKUP | | 만난사람직급 | | | Y | VARCHAR2(100) | | |
| | SALES_PHONE | | 만난사람전화 | | | Y | VARCHAR2(100) | | |

지출보고서 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|----------------|---------------------|----------|--------------|----|----|------|---------------|---------|---------------------|
| EXPENSE_REPORT | REPORT_NO | 지출보고서 | 보고서번호 | Y | | | NUMBER | | |
| | REPORT_CODE | | 보고서종류 | | Y | Y | NUMBER | | CODE_REPORT |
| | REPORT_NAME | | 보고서이름 | | | Y | VARCHAR2(100) | | |
| | EXPENSE_FIELD_CODE | | 지출분야 | | Y | Y | NUMBER | | CODE_EXPENSE_FIELD |
| | EXPENSE_PRICE | | 지출비용 | | | Y | NUMBER(20) | | |
| | EXPENSE_METHOD_CODE | | 지출수단 | | Y | Y | NUMBER | | CODE_EXPENSE_METHOD |
| | EXPENSE_DATE | | 지출일 | | | Y | DATE | | |
| | RECEIPT_NO | | 영수증번호 | | | Y | VARCHAR2(50) | | |
| | REMARKS | | 비고 | | | Y | VARCHAR2(50) | | |
| | EMP_NO | | 보고서 작성한 사원번호 | | Y | Y | NUMBER | | EMPLOYEE |

보고서 결재 정보 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|--------------------|----------------|-----------|-----------------|----|----|------|--------|---------|-----------|
| SIGN_FINISH_REPORT | SIGN_FINISH_NO | 보고서 결재 정보 | 결재완료보고서번호 | Y | | | NUMBER | | |
| | EMP_NO | | 결재한사람 | | Y | Y | NUMBER | | EMPLOYEE |
| | REPORT_CODE | | 보고서종류코드 | | | Y | NUMBER | | |
| | REPORT_NO | | 보고서번호 | | | Y | NUMBER | | |
| | SIGN_DATE | | 결재시간 | | | Y | DATE | | |
| | SIGN_CODE | | 내가 처리한 보고서 결재종류 | | Y | Y | NUMBER | | CODE_SIGN |
| | RETURN_NO | | 반려시+1되어 저장되는 값 | | | Y | NUMBER | | |

결재여부종류 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|-----------|-----------|----------|----------|----|----|------|---------------|---------|--------|
| CODE_SIGN | SIGN_CODE | 결재여부종류 | 결재여부종류코드 | Y | | | NUMBER | | |
| | SIGN_NAME | | 결재여부종류이름 | | | Y | VARCHAR2(100) | | |

지출분야 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|--------------------|--------------------|----------|---------|----|----|------|---------------|---------|--------|
| CODE_EXPENSE_FIELD | EXPENSE_FIELD_CODE | 지출분야 | 지출분야코드 | Y | | | NUMBER | | |
| | EXPENSE_FIELD_NAME | | 지출분야이름 | | | Y | VARCHAR2(100) | | |

보고서종류 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|-------------|-------------|----------|---------|----|----|------|---------------|---------|--------|
| CODE_REPORT | REPORT_CODE | 보고서종류 | 보고서종류코드 | Y | | | NUMBER | | |
| | REPORT_NAME | | 보고서종류이름 | | | Y | VARCHAR2(100) | | |

부서종류 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|-----------|----------|----------|---------|----|----|------|--------------|---------|--------|
| CODE_DEPT | DEP_NO | 부서종류 | 부서번호 | Y | | Y | NUMBER(2) | | |
| | DEP_NAME | | 부서이름 | | | | VARCHAR2(10) | | |

방문목적종류 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|-----------------|-----------------|----------|---------|----|----|------|---------------|---------|--------|
| CODE_SALES_TYPE | SALES_TYPE_CODE | 방문목적종류 | 방문목적코드 | Y | | | NUMBER | | |
| | SALES_TYPE_NAME | | 방문목적이름 | | | Y | VARCHAR2(100) | | |

지출수단 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|---------------------|---------------------|----------|---------|----|----|------|---------------|---------|--------|
| CODE_EXPENSE_METHOD | EXPENSE_METHOD_CODE | 지출수단 | 지출수단코드 | Y | | | NUMBER | | |
| | EXPENSE_METHOD_NAME | | 지출수단이름 | | | Y | VARCHAR2(100) | | |

직급종류 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|------------|------------|----------|---------|----|----|------|---------------|---------|--------|
| CODE_JIKUP | JIKUP_CODE | 직급종류 | 직급코드 | Y | | | NUMBER | | |
| | JIKUP_NAME | | 직급이름 | | | Y | VARCHAR2(100) | | |

업무분야정보 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|--------------------|----------------|----------|---------|----|----|------|---------------|---------|--------|
| CODE_BUSINESS_TYPE | BUSI_TYPE_CODE | 업무분야정보 | 업무분야코드 | Y | | | NUMBER | | |
| | BUSI_TYPE_NAME | | 업무분야이름 | | | Y | VARCHAR2(100) | | |

직원 정보 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|----------|---------------------|----------|----------|----|----|------|---------------|---------|------------|
| EMPLOYEE | EMP_NO | 직원정보테이블 | 사원번호 | Y | | Y | NUMBER | | |
| | EMP_NAME | | 사원이름 | | | | VARCHAR2(100) | | |
| | EMP_PWD | | 사원암호 | | | Y | VARCHAR2(100) | | |
| | JIKUP_CODE | | 직급코드 | | Y | | NUMBER | | CODE_JIKUP |
| | EMAIL | | 이메일 | | | | VARCHAR2(100) | | |
| | HIRE_DATE | | 입사일 | | | | DATE | SYSDATE | |
| | EMP_ROLE | | 권한 | | | | VARCHAR2(100) | 'USER' | |
| | ANNUAL_PAID_HOILDAY | | 남은연차일수 | | | | NUMBER | 15 | |
| | MGR_EMP_NO | | 직속상관 | | | | NUMBER | | |
| | DEP_NO | | 부서번호 | | Y | | NUMBER(2) | | CODE_DEPT |
| | JUMIN_NUM | | 주민번호 | | | | VARCHAR2(100) | | |
| | PHONE_NUM | | 전화번호 | | | Y | VARCHAR2(100) | | |
| | ALLOW_CODE | | 가입수락상태코드 | | Y | | NUMBER | | CODE_ALLOW |

게시판 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|----------|-----------|----------|---------|----|----|------|----------------|---------|--------|
| BOARD | B_NO | 게시판 | 게시글번호 | Y | | Y | NUMBER | | |
| | SUBJECT | | 게시글제목 | | | | VARCHAR2(1000) | | |
| | WRITER | | 작성자 | | | | VARCHAR2(100) | | |
| | REG_DATE | | 작성일 | | | Y | DATE | SYSDATE | |
| | READCOUNT | | 조회수 | | | Y | NUMBER(5) | 0 | |
| | CONTENT | | 게시글내용 | | | | NCLOB | | |

| 회의실 테이블 | | | | | | | | | |
|-------------|-----------|----------|---------|----|----|------|--------------|---------|--------|
| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
| MEETINGROOM | ROOM_CODE | 회의실 | 회의실코드 | Y | | N | VARCHAR2(10) | | |
| | ROOM_NAME | | 회의실명 | | | N | VARCHAR2(20) | | |

| 예약된 회의실 테이블 | | | | | | | | | |
|------------------|------------------|------------|-------------|----|----|------|---------------|---------|-------------|
| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
| MEETINGROOMBOARD | B_NO | 회의실 예약 게시판 | 회의실코드 | | | N | NUMBER | | |
| | EMP_NO | | 예약한 사원번호 | | Y | N | NUMBER | | EMPLOYEE |
| | DEP_NO | | 예약한 사원 부서번호 | | Y | | NUMBER | | CODE_DEPT |
| | SUBJECT | | 회의 제목 | | | | VARCHAR2(50) | | |
| | CONTENT | | 회의 내용 | | | | VARCHAR2(500) | | |
| | MIN_MEETING_TIME | | 이용 시작 시간 | | | | DATE | | |
| | MAX_MEETING_TIME | | 이용 종료 시간 | | | | DATE | | |
| | ROOM_CODE | | 예약한 회의실 코드 | Y | | | VARCHAR2(10) | | MEETINGROOM |

일정 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|----------|---------------------|----------|---------|----|----|------|--------------|---------|----------|
| CALENDAR | calendar_no | 일정 테이블 | 일정 번호 | Y | | N | Number(3) | | |
| | Emp_no | | 사원 번호 | | Y | N | Number(3) | | EMPLOYEE |
| | calendar_title | | 일정 제목 | | | N | VARCHAR2(20) | | |
| | calendar_start_date | | 일정 시작 일 | | | N | Date | | |
| | calendar_end_date | | 일정 마감 일 | | | N | Date | | |
| | Presence_or_absence | | 일정 상태 | | | N | Number(3) | | |

일정 공유 테이블

| 물리적 테이블명 | 물리적 컬럼명 | 논리적 테이블명 | 논리적 컬럼명 | PK | FK | NULL | TYPE | DEFAULT | 참조 테이블 |
|---------------|--------------|----------|----------|----|----|------|---------------|---------|----------|
| scheduleshare | Share_no | 일정 공유 | 공유 일정 번호 | Y | | N | Number(9) | | |
| | thesender | | 보낸 이 | | | N | Number(9) | | |
| | therecipient | | 받는 이 | | | N | Number(9) | | |
| | calendar_no | | 일정 번호 | | Y | | Number(9) | | CALENDAR |
| | subject | | 제목 | | | N | VARCHAR2(200) | | |
| | timesent | | 보낸 시각 | | | | Date | SYSDATE | |

일일 업무 보고서

| Class & Interface | Description |
|-------------------------|--|
| DailyBusinessController | 클라이언트 요청을 처리한 후 일일업무보고서 JSP에서 지정된 뷰에 모델 객체를 넘겨주는 역할을 하는 클래스 |
| DailyService | DailyServiceImpl에서 사용할 메소드 생성한 인터페이스 |
| DailyServiceImpl | DailyService 인터페이스를 구현한 클래스로 DailyBusinessController에서 데이터를 받아 비즈니스 로직을 처리하고 DB연동이 필요한 경우 해당 처리를 하는 DailyDAO를 호출하는 클래스 |
| DailyDAO | 일일업무보고서의 입력/수정/삭제/검색에 관한 DB연동 수행하는 인터페이스 |
| DailyDAOImpl | DailyDAO 인터페이스를 구현하고 일일업무보고서의 입력/수정/삭제/검색에 관한 DB연동 수행하는 클래스 |
| DailyBusiDTO | 1개의 일일 업무 보고서의 상세정보 데이터를 담아 이동하는 getter/setter 메소드 선언한 클래스 |
| DailyBusiSearchDTO | 일일업무보고서 검색관련 데이터를 담아 이동하는 getter/setter 메소드 선언한 클래스 |
| DailyBusiUtil | 일일업무보고서 페이지 처리를 하기 위해 선언한 클래스 |
| Next_App_Emp_List | 다음 결재자 목록 데이터를 담아 이동하는 getter/setter 메소드 선언한 클래스 |

영업 업무 보고서

| Class & Interface | Description |
|-------------------|--|
| SalesController | 클라이언트 요청을 처리한 후 영업업무보고서 JSP에서 지정된 뷰에 모델 객체를 넘겨주는 역할하는 클래스 |
| SalesService | SalesServiceImpl에서 사용할 메소드 생성한 인터페이스 |
| SalesServiceImpl | SalesService 인터페이스를 구현한 클래스로 SalesController에서 데이터를 받아 비즈니스 로직을 처리하고 DB연동이 필요한 경우 해당 처리를 하는 SalesDAO를 호출하는 클래스 |
| SalesDAO | 영업업무보고서의 입력/수정/삭제/검색에 관한 DB연동 수행하는 인터페이스 |
| SalesDAOImpl | DailyDAO 인터페이스를 구현하고 일일업무보고서의 입력/수정/삭제/검색에 관한 DB연동 수행하는 클래스 |
| SalesDTO | 1개의 일일업무보고서의 상세정보 데이터를 담아 이동하는 getter/setter 메소드 선언한 클래스 |
| SalesSearchDTO | 일일업무보고서 검색관련 데이터를 담아 이동하는 getter/setter 메소드 선언한 클래스 |
| SalesUtil | 일일업무보고서 페이지 처리를 하기 위해 선언한 클래스 |
| Next_App_Emp_List | 다음 결재자 목록 데이터를 담아 이동하는 getter/setter 메소드 선언한 클래스 |

지출 보고서

| Class & Interface | Description |
|--------------------|--|
| ExpenseController | 클라이언트 요청을 처리한 후 지출보고서 JSP에서 지정된 뷰에 모델 객체를 넘겨주는 역할하는 클래스 |
| ExpenseService | ExpenseServiceImpl에서 사용할 메소드 생성한 인터페이스 |
| ExpenseServiceImpl | ExpenseService 인터페이스를 구현한 클래스로 ExpenseController에서 데이터를 받아 비즈니스 로직을 처리하고 DB연동이 필요한 경우 해당 처리를 하는 ExpenseDAO를 호출하는 클래스 |
| ExpenseDAO | 지출보고서의 입력/수정/삭제/검색에 관한 DB연동 수행하는 인터페이스 |
| ExpenseDAOImpl | ExpenseDAO 인터페이스를 구현하고 지출보고서의 입력/수정/삭제/검색에 관한 DB연동 수행하는 클래스 |
| ExpenseDTO | 1개의 지출보고서의 상세정보 데이터를 담아 이동하는 getter/setter 메소드 선언한 클래스 |
| ExpenseSearchDTO | 지출보고서 검색관련 데이터를 담아 이동하는 getter/setter 메소드 선언한 클래스 |
| ExpenseUtil | 지출보고서 페이지 처리를 하기 위해 선언한 클래스 |
| Next_App_Emp_List | 다음 결재자 목록 데이터를 담아 이동하는 getter/setter 메소드 선언한 클래스 |

Employee

| Class & Interface | Description |
|---------------------|---|
| EmployeeController | 가상주소 /empList.do, /empDetailForm.do, /empRegForm.do, 로 접속하면 호출되는 메소드 선언 ModelAndView 객체에 호출 jsp 페이지명과 응답물을 저장하고 리턴 |
| EmployeeDAO | EmployeeDAOImpl에서 사용할 메소드 선언 |
| EmployeeDAOImpl | 관리자페이지의 입력 / 수정 / 삭제 / 검색에 관한 EmployeeDAO의 메소드를 오버라이딩 |
| EmployeeService | EmployeeServiceImpl에서 사용할 메소드 선언 |
| EmployeeServiceImpl | 관리자페이지의 입력 / 수정 / 삭제 / 검색에 관한 EmployeeService의 메소드를 오버라이딩 |
| EmployeeDTO | 1개의 직원정보의 상세정보 데이터를 담아 이동하는 getter/setter 선언 |
| EmployeeSearchDTO | 관리자페이지 검색관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |

마이페이지

| Class & Interface | Description |
|-------------------|--|
| MypageController | 가상주소 /mypageMain.do 로 접속하면 호출되는 메소드 선언 ModelAndView 객체에 호출 jsp 페이지명과 응답물을 저장하고 리턴 |
| MypageDAO | MypageDAOImpl에서 사용할 메소드 선언 |
| MypageDAOImpl | 내 정보의 입력 / 수정 / 삭제 / 검색에 관한 MypageDAO의 메소드를 오버라이딩 |
| MypageService | MypageServiceImpl에서 사용할 메소드 선언 |
| MypageServiceImpl | 내 정보의 입력 / 수정 / 삭제 / 검색에 관한 MypageService의 메소드를 오버라이딩 |

공지사항

| Class & Interface | Description |
|-------------------|--|
| BoardController | 가상주소 /boardList.do, /boardRegForm.do, /boardDetailForm.do, /boardUpDelForm.do 로 접속하면 호출되는 메소드 선언 ModelAndView 객체에 호출 jsp 페이지명과 응답물을 저장하고 리턴 |
| BoardDAO | BoardDAOImpl에서 사용할 메소드 선언 |
| BoardDAOImpl | 공지사항페이지의 입력 / 수정 / 삭제 / 검색에 관한 BoardDAO의 메소드를 오버라이딩 |
| BoardService | BoardServiceImpl에서 사용할 메소드 선언 |
| BoardServiceImpl | 공지사항페이지의 입력 / 수정 / 삭제 / 검색에 관한 BoardService의 메소드를 오버라이딩 |
| BoardDTO | 1개의 공지사항의 상세정보 데이터를 담아 이동하는 getter/setter 선언 |
| BoardSearchDTO | 공지사항페이지의 검색관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |
| BoardValidator | 공지사항 입력/수정시 유효성 체크를 위한 메소드 선언 |
| Util.java | 페이지 처리를 위한 메소드 선언 |

회의실 예약

| Class & Interface | Description |
|--------------------|----------------------------------|
| MeetingController | 회의실 예약 관련 가상주소로 접속하면 호출되는 메소드 선언 |
| MeetingDAO | MeetingDAOImpl에서 사용할 메소드 선언 |
| MeetingDAOImpl | MeetingDAO 의 메소드들을 오버라이딩 |
| MeetingService | MeetingServiceImpl에서 사용할 메소드 선언 |
| MeetingServiceImpl | MeetingService의 메소드들 오버라이딩 |
| MBoardSearchDTO | 예약된 게시판 검색관련 데이터를 전송하는 DTO 객체 |

일정관리

| Class & Interface | Description |
|---------------------|---|
| ScheduleController | ScheduleServiceImpl 메소드를 호출하여 일정관리 관련 DB연동 결과물을 가져오고 화면 출력을 위한 JSP 페이지명과 ModelAndView 객체에 저장하고 리턴 |
| ScheduleService | ScheduleServiceImpl에서 사용할 메소드 선언 |
| ScheduleServiceImpl | 일정관리의 입력 / 수정 / 삭제 / 검색에 관한 ScheduleService의 메소드를 오버라이딩 한 후 ScheduleDAOImpl 메소드 호출 |
| ScheduleDAO | ScheduleServiceImpl에서 사용할 메소드 선언 |
| ScheduleDAOImpl | 일정관리의 입력 / 수정 / 삭제 / 검색에 관한 ScheduleDAO의 메소드를 오버라이딩 한 후 DB 연동 결과를 리턴 |
| ScheduleDTO | 일정관리 등록 시 상세정보 데이터를 담아 이동하는 getter/setter 선언 |
| NotificationDTO | 일정관리의 알림 목록 상세정보 데이터를 담아 이동하는 getter/setter 선언 |

로그인

| Class & Interface | Description |
|-------------------|---|
| LoginController | LoginServiceImpl 메소드를 호출하여 로그인 관련 DB연동 결과물을 가져오고 화면 출력을 위한 JSP 페이지명과 ModelAndView 객체에 저장하고 리턴 |
| LoginService | LoginServiceImpl에서 사용할 메소드 선언 |
| LoginServiceImpl | 로그인의 입력 / 수정 / 검색에 관한 LoginService의 메소드를 오버라이딩 한 후 LoginDAOImpl 메소드 호출 |
| LoginDAO | LoginDAOImpl에서 사용할 메소드 선언 |
| LoginDAOImpl | 로그인의 입력 / 수정 / 검색에 관한 LoginDAO의 메소드를 오버라이딩 한 후 DB 연동 결과를 리턴 |

회원가입

| Class & Interface | Description |
|-------------------|---|
| JoinController | JoinServiceImpl 메소드를 호출하여 회원가입 관련 DB연동 결과물을 가져오고 화면 출력을 위한 JSP 페이지명과 ModelAndView 객체에 저장하고 리턴 |
| JoinService | JoinServiceImpl에서 사용할 메소드 선언 |
| JoinServiceImpl | 회원가입의 입력 / 검색에 관한 JoinService의 메소드를 오버라이딩 한 후 JoinDAOImpl 메소드 호출 |
| JoinDAO | JoinDAOImpl에서 사용할 메소드 선언 |
| JoinDAOImpl | 회원가입의 입력 / 검색에 관한 JoinDAO의 메소드를 오버라이딩 한 후 DB 연동 결과를 리턴 |
| JoinDTO | 회원가입 등록 시 상세정보 데이터를 담아 이동하는 getter/setter 선언 |

일일 업무/영업 업무/지출 보고서

| JSP PAGE | Description |
|----------------------|------------------------------|
| dailyListForm | 일일업무보고서 검색 화면 및 목록을 보여주는 JSP |
| dailybusinessRegForm | 일일업무보고서 등록 화면을 보여주는 JSP |
| dailybusiDetailForm | 1개의 일일업무보고서 상세 화면을 보여주는 JSP |
| salesListForm | 영업업무보고서 검색 화면 및 목록을 보여주는 JSP |
| salesRegForm | 영업업무보고서 등록 화면을 보여주는 JSP |
| salesDetailForm | 영업업무보고서 검색 화면 및 목록을 보여주는 JSP |
| expenseListForm | 지출보고서 검색 화면 및 목록을 보여주는 JSP |
| expenseRegForm | 지출보고서 등록 화면을 보여주는 JSP |
| expenseDetailForm | 지출보고서 검색 화면 및 목록을 보여주는 JSP |

회의실 예약

| JSP PAGE | Description |
|-------------------|--------------------------------|
| meetingBoardList | 예약된 회의실 게시판을 보여주는 jsp 페이지 |
| meetingDetailForm | 예약된 회의실 게시판 상세화면을 보여주는 jsp 페이지 |
| meetingResForm | 회의실 예약 화면을 보여주는 jsp페이지 |
| meetingUoDelForm | 예약된 회의실 게시글 수정하는 jsp 페이지 |

관리자 가입 관리 & 마이페이지

| JSP PAGE | Description |
|-------------------|---|
| empList.jsp | 직원관리 및 가입관리 검색화면을 보여주는 jsp 페이지 |
| empDetailForm.jsp | 직원 상세 정보를 보여주는 jsp 페이지 |
| empRegForm.jsp | 직원관리 등록 화면을 보여주는 jsp 페이지 |
| empUpDelForm.jsp | 직원관리 수정/삭제 화면을 보여주는 jsp 페이지 |
| moveData_emp.jsp | 상세보기, 글쓰기 할 때 가져갈 검색결과, 선택한 페이지 번호, 보여질 행의 개수를 저장하는 jsp 페이지 |
| mypageMain.jsp | 마이페이지 화면을 보여주는 jsp 페이지 |
| mypageUpdate.jsp | 마이페이지 수정 화면을 보여주는 jsp 페이지 |

일정관리 및 로그인

| JSP PAGE | Description |
|--------------------|------------------------------------|
| calDetail.jsp | 일정관리 상세보기 팝업창을 띄우는 jsp 페이지 |
| calRegForm.jsp | 일정관리 등록 팝업창을 띄우는 jsp 페이지 |
| calSharing.jsp | 일정공유 팝업창을 띄우는 jsp페이지 |
| planList.jsp | 일정관리 화면을 보여주는 jsp 페이지 |
| newsDetailForm.jsp | 일정관리 화면에서 알림 상세보기 팝업창을 띄우는 jsp 페이지 |
| loginForm.jsp | 로그인 화면을 보여주는 jsp 페이지 |
| idFindForm.jsp | 아이디 찾기 화면을 보여주는 jsp 페이지 |
| pwdFindForm.jsp | 비밀번호 찾기 화면을 보여주는 jsp 페이지 |
| pwdChangeForm.jsp | 비밀번호 변경 화면을 보여주는 jsp 페이지 |
| joinForm.jsp | 회원가입 화면을 보여주는 jsp 페이지 |
| idView.jsp | 가입된 아이디를 보여주는 팝업창을 띄우는 jsp 페이지 |

일일 업무 보고서

| Class & Interface | Method | 기능 |
|--|---|--|
| DailyBusinessController (6개의 메소드로 구성) | ModelAndView godailyListForm(HttpServletRequest session, DailyBusiSearchDTO dailyBusiSearchDTO) | 가상주소 /dailyListForm.do 로 접근하면 호출되는 메소드 선언 |
| | ModelAndView godailyRegForm(HttpServletRequest session, DailyBusiDTO dailyBusiDTO) | 가상주소 /dailybusinessRegForm.do 로 접근하면 호출되는 메소드 선언 |
| | int insertDailBusiReport(HttpServletRequest session, DailyBusiDTO dailyBusiDTO) | 가상주소 /dailyBusiRegProc.do 로 접근하면 [1개의 일일업무보고서 입력 적용행의 개수]를 리턴하는 메소드 선언 |
| | public ModelAndView dailyDetailForm (HttpServletRequest session ,@RequestParam(value="report_no") int report_no ,@RequestParam(value="report_no") int report_code) | 가상주소 /dailyBusinessDetailForm.do 로 접근하면 호출되는 메소드 선언 |
| | int alterDailBusiReport(HttpServletRequest session, DailyBusiDTO dailyBusiDTO) | 가상주소 /dailyAlterProc.do 로 접근하면 호출되는 메소드 선언 |
| | int signDailBusiReport(HttpServletRequest session, DailyBusiDTO dailyBusiDTO) | 가상주소 /dailyUpdelProc.do 로 접근하면 호출되는 메소드 선언 |
| DailyService | DailyServiceImpl 클래스의 메소드와 동일 | 인터페이스로 DailyServiceImpl 의 메소드 목록을 규정 |
| DailyServiceimpl (7개의 메소드로 구성) | List<Map<String, String>> getDailyList(DailyBusiSearchDTO dailyBusiSearchDTO); | 검색한 일일업무보고서 목록을 리턴하는 메소드 선언 |
| | int first_insertDailyReport(DailyBusiDTO dailyBusiDTO); | [1개의 일일업무보고서 입력 후 적용행의 개수]를 리턴하는 메소드 선언 |
| | int signDailBusiReport(DailyBusiDTO dailyBusiDTO); | 1개의 일일업무보고서의 결재 후 적용행의 개수]를 리턴하는 메소드 선언 |
| | int signMinusBusiReport(DailyBusiDTO dailyBusiDTO); | 1개의 일일업무보고서의 결재 후 적용행의 개수]를 리턴하는 메소드 선언 |
| | List<Next_App_Emp_List> get_next_app_list(DailyBusiDTO dailyBusiDTO); | 다음 결재자 목록 리스트 리턴하는 메소드 선언 |
| | DailyBusiDTO getDaily(int report_no,int emp_no); | 선택한 일일업무보고서 상세정보를 담아오는 메소드 선언 |
| | int alterDailBusiReport(DailyBusiDTO dailyBusiDTO); | 반려된 일일업무보고서 수정 후 적용행의 개수를 리턴하는 메소드 선언 |

일일 업무 보고서

| Class & Interface | Method | 기능 |
|--------------------------------|--|--|
| DailyDAO | DailyDAOImpl 클래스의 메소드와 동일 | 인터페이스로 DailyDAOImpl 의 메소드 목록을 규정 |
| | List<Map<String, String>> getDailyList(DailyBusiSearchDTO dailyBusiSearchDTO); | [검색한 일일업무보고서 목록]을 리턴하는 메소드 선언 |
| | Map<String, String> getLastAppMan(DailyBusiSearchDTO dailyBusiSearchDTO); | 최종 결제자의 정보를 얻어오는 메소드 선언 |
| | String first_app_getSign_date(DailyBusiSearchDTO dailyBusiSearchDTO); | 만약 첫 결재자면 바로 앞 결재자의 시간을 얻어오는 메소드 선언 |
| | String next_app_getSign_date(DailyBusiSearchDTO dailyBusiSearchDTO); | 다음결제자의 결제시간을 얻어오는 메소드 선언 |
| | int getdailyBusiListAllTotCnt(DailyBusiSearchDTO dailyBusiSearchDTO); | 일일업무보고서 총 개수 구하는 메소드 선언 |
| | int getdailyBusiListTotCnt(DailyBusiSearchDTO dailyBusiSearchDTO); | 일일업무보고서 검색 결과물의 총 개수 구하는 메소드 선언 |
| | String getSign_date(DailyBusiSearchDTO dailyBusiSearchDTO); | 결재여부 날짜 리턴 받는 메소드 선언 |
| | String getSign_code(DailyBusiSearchDTO dailyBusiSearchDTO); | 결재여부 코드 리턴 받는 메소드 선언 |
| | String next_app_getSign_code(DailyBusiSearchDTO dailyBusiSearchDTO); | 다음 결재자 결재여부코드를 가져오는 메소드 선언 |
| | int first_insertDailyReport(DailyBusiDTO dailyBusiDTO); | 자신이 작성한 일일업무보고서 적용 행의 개수 리턴 메소드 선언 |
| DailyDAOImpl (35개의 메소드로 구성) | int first_insertSignFinishReport_Sign_emp(DailyBusiDTO dailyBusiDTO); | 자신의 일일업무보고서의 다음 결재자를 보고서결재정보 테이블에 적용하는 메소드 선언 |
| | Map<String, String> getLastAppMan(DailyBusiDTO dailyBusiDTO); | 최종 결제자의 정보를 얻어오는 메소드 선언 |
| | DailyBusiDTO getBoard(Map<String, String> getBoardData); | 일일업무보고서 번호에 해당하는 상세화면에 보여주는 데이터 검색 결과 개수 리턴 메소드 선언 |
| | List<Next_App_Emp_List> get_next_app_list2(int emp_no); | 상세화면 다음 결재자 목록 리스트 리턴하는 메소드 선언 |
| | int updateSignCode(DailyBusiDTO dailyBusiDTO); | 결재/반려에서 결재코드 업데이트 성공 여부 리턴 메소드 선언 |
| | int new_insertSignFinishReport_Sign_emp(DailyBusiDTO dailyBusiDTO); | 결재/반려에서 새로운 결재코드 입력 행의 개수 리턴 메소드 선언 |
| | int updateSignCodeMinus(DailyBusiDTO dailyBusiDTO); | 결재/반려에서 결재코드 반려로 업데이트 성공 여부 리턴 메소드 선언 |
| | int insertReturnReport(DailyBusiDTO dailyBusiDTO); | 결재/반려에서 결재코드 반려로 업데이트 성공 시 반려테이블에 반려된 보고서번호와 반려사유 입력 행의 개수 리턴 메소드 선언 |
| | List<Next_App_Emp_List> get_next_app_list(DailyBusiDTO dailyBusiDTO); | 등록화면 다음 결재자 목록 리스트 리턴하는 메소드 선언 |
| | int alterDailBusiContent(DailyBusiDTO dailyBusiDTO); | 들어온 번호를 통해 컨텐츠(내용) 업데이트 하는 메소드 선언 |

일일 업무 보고서

| Class & Interface | Method | 기능 |
|--------------------------------|--|--|
| DailyDAOImpl (35개의 메소드로 구성) | int alterDeleteSignDate(DailyBusiDTO dailyBusiDTO); | 일일업무보고서를 수정 시 다음 결재자들의 결재 날짜를 null로 업데이트하는 메소드 선언 |
| | int alterNextAppUpdateSignCode(DailyBusiDTO dailyBusiDTO); | 들어온 번호를 통해 다음 결재자의 sign_code 업데이트 하는 메소드 선언 |
| | Map<String, String> getNextSign_finish_report(DailyBusiDTO dailyBusiDTO); | 일일업무보고서가 수정했을 경우 다음 결재자 emp_no 데이터를 리턴하는 메소드 선언 |
| | int alter_Next_App_Sign_Code(DailyBusiDTO dailyBusiDTO); | 일일업무보고서가 수정했을 경우 다음 결재자 sign_code를 -1로 업데이트 후 리턴 받는 메소드 선언 |
| | int alter_Next_App_Sign_Date(DailyBusiDTO dailyBusiDTO); | 일일업무보고서가 수정했을 경우 다음 결재자 sign_date를 현재로 업데이트 하는 메소드 선언 |
| | Map<String, String> getNextAppMan(DailyBusiDTO dailyBusiDTO); | 일일업무보고서의 다음 결제자의 정보를 얻어오는 메소드 선언 |
| | Map<String, String> getLogin_NextAppMan(DailyBusiDTO dailyBusiDTO); | 로그인한 사람의 다음 결제자의 정보를 얻어오는 메소드 선언 |
| | int app_setTime(DailyBusiDTO dailyBusiDTO); | 다음 결재자 없으면 현 결재자의 시간을 업데이트 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | List<Map<String, String>> getAppList(DailyBusiDTO dailyBusiDTO); | 일일업무보고서의 해당 번호의 결재 내역 데이터를 리턴받는 메소드 선언 |
| | String next_appList_getSign_date(Map<String, String> myInfo); | 일일업무보고서의 다음 결재자의 날짜 리턴받는 메소드 선언 |
| | String my_appList_getSign_date(Map<String, String> myInfo); | 로그인한 사원의 일일업무보고서 등록한 날짜 리턴받는 메소드 선언 |
| | int alterDeleteReturnReport(DailyBusiDTO dailyBusiDTO); | 반려된 일일업무보고서 수정 시 반려테이블의 반려사유 null로 업데이트 후 리턴 받는 메소드 선언 |
| | String nextAppManEmp_no(Map<String, String> AppEmp_no); | 반려된 일일업무보고서의 다음 결재자의 사원번호를 리턴 받는 메소드 선언 |
| | int haveOrNoneReturn_report(DailyBusiDTO dailyBusiDTO); | 반려된 해당 일일업무보고서의 개수를 리턴 받는 메소드 선언 |
| | int insertReturnResult(DailyBusiDTO dailyBusiDTO); | 결재/반려에서 결재코드 반려로 업데이트 성공 시 반려테이블에 반려된 보고서번호와 반려사유 입력 행의 개수 리턴 메소드 선언 |
| DailyBusiDTO | getter/setter 메소드로 구성 | [일일업무보고서]의 입력/수정/결재/반려 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |
| DailyBusiSearchDTO | getter/setter 메소드로 구성 | [일일업무보고서]의 검색 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |
| DailyBusiUtil | public static Map<String, Integer> getPagingMap(int selectPageNo, int rowCntPerPage, int totCnt) | [일일업무보고서]의 페이지 관련 데이터를 리턴하는 메소드 선언 |

영업 업무 보고서

| Class & Interface | Method | 기능 |
|-----------------------------------|---|--|
| SalesController | ModelAndView goSalesListForm(HttpServletRequest session, SalesSearchDTO salesSearchDTO) | 가상주소 /salesListForm.do 로 접근하면 호출되는 메소드 선언 |
| | ModelAndView goSalesRegForm(HttpServletRequest session, SalesDTO salesDTO) | 가상주소 /salesRegForm.do 로 접근하면 호출되는 메소드 선언 |
| | int insertSalesReport(HttpServletRequest session, SalesDTO salesDTO) | 가상주소 /salesRegProc.do 로 접근하면 [1개의 영업업무보고서 입력 적용행의 개수]를 리턴하는 메소드 선언 |
| | public ModelAndView boardDetailForm(@RequestParam(value="report_no") int report_no, @RequestParam(value="report_code") int report_code, HttpServletRequest session) | 가상주소 /salesDetailForm.do 로 접근하면 호출되는 메소드 선언 |
| | Int signDailBusiReport(HttpServletRequest session, SalesDTO salesDTO) | 가상주소 /salesUpdelProc.do 로 접근하면 호출되는 메소드 선언 |
| | int alterDailBusiReport(HttpServletRequest session, SalesDTO salesDTO) | 가상주소 /salesAlterProc.do 로 접근하면 호출되는 메소드 선언 |
| SalesService | SalesServiceImpl 클래스의 메소드와 동일 | 인터페이스로 SalesServiceImpl 의 메소드 목록을 규정 |
| SalesServiceImpl (7개의 메소드로 구성) | int first_insertSalesReport(SalesDTO salesDTO); | 검색한 영업업무보고서 목록을 리턴하는 메소드 선언 |
| | List<Map<String, String>> getSalesList(SalesSearchDTO salesSearchDTO); | [1개의 영업업무보고서 입력 후 적용행의 개수]를 리턴하는 메소드 선언 |
| | SalesDTO getBoard(int report_no, int emp_no2, int report_code); | 1개의 영업업무보고서의 결재 후 적용행의 개수]를 리턴하는 메소드 선언 |
| | int signDailBusiReport(SalesDTO salesDTO); | 1개의 영업업무보고서의 결재 후 적용행의 개수]를 리턴하는 메소드 선언 |
| | int signMinusBusiReport(SalesDTO salesDTO); | 다음 결재자 목록 리스트 리턴하는 메소드 선언 |
| | int alterDailBusiReport(SalesDTO salesDTO); | 선택한 영업업무보고서 상세정보를 담아오는 메소드 선언 |
| | List<Next_App_Emp_List> get_next_app_list(SalesDTO salesDTO); | 반려된 영업업무보고서 수정 후 적용행의 개수를 리턴하는 메소드 선언 |

영업 업무 보고서

| Class & Interface | Method | 기능 |
|--------------------------------|--|--|
| SalesDAO | SalesDAOImpl 클래스의 메소드와 동일 | 인터페이스로 SalesDAOImpl 의 메소드 목록을 규정 |
| SalesDAOImpl (34개의 메소드로 구성) | List<Map<String, String>> getSalesList(SalesSearchDTO salesSearchDTO); | [검색한 영업업무보고서 목록]을 리턴하는 메소드 선언 |
| | Map<String, String> getLastAppMan(SalesSearchDTO salesSearchDTO); | 최종 결제자의 정보를 얻어오는 메소드 선언 |
| | String first_app_getSign_date(SalesSearchDTO salesSearchDTO); | 만약 첫 결재자면 바로 앞 결재자의 시간을 얻어오는 메소드 선언 |
| | String next_app_getSign_date(SalesSearchDTO salesSearchDTO); | 다음결제자의 결제시간을 얻어오는 메소드 선언 |
| | int getsalesListAllTotCnt(SalesSearchDTO salesSearchDTO); | 영업업무보고서 총 개수 구하는 메소드 선언 |
| | int getsalesListTotCnt(SalesSearchDTO salesSearchDTO); | 영업업무보고서 검색 결과물의 총 개수 구하는 메소드 선언 |
| | String getSign_code(SalesSearchDTO salesSearchDTO); | 결재여부 코드 리턴 받는 메소드 선언 |
| | String next_app_getSign_code(SalesSearchDTO salesSearchDTO); | 다음 결재자 결재여부코드를 가져오는 메소드 선언 |
| | int first_insertSalesReport(SalesDTO salesDTO); | 자신이 작성한 영업업무보고서 적용 행의 개수 리턴 메소드 선언 |
| | int first_insertSignFinishReport_Sign_emp(SalesDTO salesDTO); | 자신의 영업업무보고서의 다음 결재자를 보고서결재정보 테이블에 적용하는 메소드 선언 |
| | Map<String, String> getLastAppMan(SalesDTO salesDTO); | 최종 결제자의 정보를 얻어오는 메소드 선언 |
| | SalesDTO getBoard(Map<String, String> getBoardData); | 영업업무보고서 번호에 해당하는 상세화면에 보여주는 데이터 검색 결과 개수 리턴 메소드 선언 |
| | List<Next_App_Emp_List> get_next_app_list2(int emp_no2); | 상세화면 다음 결재자 목록 리스트 리턴하는 메소드 선언 |
| | int updateSignCode(SalesDTO salesDTO); | 결재/반려에서 결재코드 업데이트 성공 여부 리턴 메소드 선언 |
| SalesDAOImpl (34개의 메소드로 구성) | int new_insertSignFinishReport_Sign_emp(SalesDTO salesDTO); | 결재/반려에서 새로운 결재코드 입력 행의 개수 리턴 메소드 선언 |
| | int updateSignCodeMinus(SalesDTO salesDTO); | 결재/반려에서 결재코드 반려로 업데이트 성공 여부 리턴 메소드 선언 |
| | int insertReturnReport(SalesDTO salesDTO); | 결재/반려에서 결재코드 반려로 업데이트 성공 시 반려테이블에 반려된 보고서번호와 반려사유 입력 행의 개수 리턴 메소드 선언 |
| | List<Next_App_Emp_List> get_next_app_list(SalesDTO salesDTO); | 등록화면 다음 결재자 목록 리스트 리턴하는 메소드 선언 |
| | int alterSalesContent(SalesDTO salesDTO); | 들어온 번호를 통해 컨텐츠(내용) 업데이트 하는 메소드 선언 |

영업 업무 보고서

| Class & Interface | Method | 기능 |
|--------------------------------|--|--|
| SalesDAOImpl (34개의 메소드로 구성) | int alterDeleteSignDate(SalesDTO salesDTO); | 영업업무보고서를 수정 시 다음 결재자들의 결재 날짜를 null로 업데이트하는 메소드 선언 |
| | int alterNextAppUpdateSignCode(SalesDTO salesDTO); | 들어온 번호를 통해 다음 결재자의 sign_code 업데이트 하는 메소드 선언 |
| | Map<String, String> getNextSign_finish_report(SalesDTO salesDTO); | 영업업무보고서가 수정했을 경우 다음 결재자 emp_no 데이터를 리턴하는 메소드 선언 |
| | int alter_Next_App_Sign_Code(SalesDTO salesDTO); | 영업업무보고서가 수정했을 경우 다음 결재자 sign_code를 -1로 업데이트 후 리턴 받는 메소드 선언 |
| | int alter_Next_App_Sign_Date(SalesDTO salesDTO); | 영업업무보고서가 수정했을 경우 다음 결재자 sign_date를 현재로 업데이트 하는 메소드 선언 |
| | Map<String, String> getNextAppMan(SalesDTO salesDTO); | 영업업무보고서의 다음 결제자의 정보를 얻어오는 메소드 선언 |
| | Map<String, String> getLogin_NextAppMan(SalesDTO salesDTO); | 로그인한 사람의 다음 결제자의 정보를 얻어오는 메소드 선언 |
| | int app_setTime(SalesDTO salesDTO); | 다음 결재자 없으면 현 결재자의 시간을 업데이트 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | List<Map<String, String>> getAppList(SalesDTO salesDTO); | 영업업무보고서의 해당 번호의 결재 내역 데이터를 리턴받는 메소드 선언 |
| | String next_appList_getSign_date(Map<String, String> myInfo); | 영업업무보고서의 다음 결재자의 날짜 리턴받는 메소드 선언 |
| | String my_appList_getSign_date(Map<String, String> myInfo); | 로그인한 사원의 영업업무보고서 등록한 날짜 리턴받는 메소드 선언 |
| | int alterDeleteReturnReport(SalesDTO salesDTO); | 반려된 영업업무보고서 수정 시 반려테이블의 반려사유 null로 업데이트 후 리턴 받는 메소드 선언 |
| | String nextAppManEmp_no(Map<String, String> AppEmp_no); | 반려된 영업업무보고서의 다음 결재자의 사원번호를 리턴 받는 메소드 선언 |
| | int haveOrNoneReturn_report(SalesDTO salesDTO); | 반려된 해당 영업업무보고서의 개수를 리턴 받는 메소드 선언 |
| | int insertReturnResult(SalesDTO salesDTO); | 결재/반려에서 결재코드 반려로 업데이트 성공 시 반려테이블에 반려된 보고서번호와 반려사유 입력 행의 개수 리턴 메소드 선언 |
| SalesDTO | getter/setter 메소드로 구성 | [영업업무보고서]의 입력/수정/결재/반려 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |
| SalesSearchDTO | getter/setter 메소드로 구성 | [영업업무보고서]의 검색 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |
| SalesUtil | public static Map<String, Integer> getPagingMap(int selectPageNo, int rowCntPerPage, int totCnt) | [영업업무보고서]의 페이징 관련 데이터를 리턴하는 메소드 선언 |

지출 보고서

| Class & Interface | Method | 기능 |
|--------------------|---|--|
| ExpenseController | ModelAndView goExpenseListForm(HttpServletRequest session, ExpenseSearchDTO expenseSearchDTO) | 가상주소 /expenseListForm.do 로 접근하면 호출되는 메소드 선언 |
| | ModelAndView goExpenseRegForm(HttpServletRequest session, ExpenseDTO expenseDTO) | 지출보고서 등록을 누르면 호출되는 메소드 선언 |
| | int insertExpenseReport(HttpServletRequest session, ExpenseDTO expenseDTO) | 가상주소 /expenseRegProc.do 로 접근하면 [1개의 일일업무보고서 입력 적용행의 개수]를 리턴하는 메소드 선언 |
| | public ModelAndView boardDetailForm(@RequestParam(value="report_no") int report_no ,@RequestParam(value="report_code") int report_code ,HttpServletRequest session) | 지출보고서 상세보기를 누르면 호출되는 메소드 선언 |
| | int signExpenseReport(HttpServletRequest session, ExpenseDTO expenseDTO) | 결재 또는 반려 시 호출되는 메소드 선언 |
| | int alterExpenseReport(HttpServletRequest session, ExpenseDTO expenseDTO) | 수정 시 호출되는 메소드 선언 |
| ExpenseService | ExpenseServiceImpl 메소드와 동일 | 인터페이스로 ExpenseServiceImpl 의 메소드 목록을 규정 |
| ExpenseServiceImpl | public int first_insertExpenseReport(ExpenseDTO expenseDTO) | [1개의 지출보고서 입력 후 적용행의 개수]를 리턴하는 메소드 선언 |
| | PublicList<Map<String, String>> getExpenseList(ExpenseSearchDTO expenseSearchDTO) | [검색한 지출보고서 목록]을 리턴하는 메소드 선언 |
| | public ExpenseDTO getBoard(int report_no, int emp_no2, int report_code) | [1개의 지출보고서 글]을 리턴하는 메소드 선언 |
| | public int signExpenseReport(ExpenseDTO expenseDTO) | 결재를 눌렀을 경우 일어나는 메소드 정의 |

지출 보고서

| Class & Interface | Method | 기능 |
|--------------------|---|---|
| ExpenseServiceImpl | public int signMinusExpenseReport(ExpenseDTO expenseDTO) | 반려를 눌렀을 경우 적용되는 행의 개수를 리턴하는 메소드선언 |
| | public int alterExpenseReport(ExpenseDTO expenseDTO) | 수정을 눌렀을 경우 적용되는 행의 개수를 리턴하는 |
| ExpenseDAO | ExpenseDAOImpl 메소드와 동일 | 인터페이스로 ExpenseDAOImpl 메소드를 규정 |
| ExpenseDAOImpl | public int first_insertExpenseReport(ExpenseDTO expenseDTO) | 첫 지출보고서 등록시 실행 후 입력되는 적용행의 개수를 리턴하는 메소드 |
| | public int first_insertSignFinishReport_Sign_emp(ExpenseDTO expenseDTO) | 첫 지출보고서 등록시 다음결재자의 정보를 입력 후 적용행의 개수를 리턴하는 메소드 |
| | public List<Map<String, String>> getExpenseList(ExpenseDTO expenseDTO) | 지출보고서 목록을 리턴하는 메소드 |
| | public ExpenseDTO getBoard(Map<String, String> getBoardData) | 한 개의 지출보고서 목록 글의 정보를 리턴하는 메소드 |
| | public Map<String, String> getLastAppMan(ExpenseDTO expenseDTO) | 최종결재자의 정보를 얻어오는 메소드 |
| | public Map<String, String> getNextAppMan(ExpenseDTO expenseDTO) | 지출보고서의 다음 결제자의 정보를 얻어오는 메소드 선언 |
| | public int updateSignCode(ExpenseDTO expenseDTO) | 사인코드 업데이트후 적용행의 개수를 리턴하는 메소드 |
| | PublicMap<String, String>getNextSign_finish_report(ExpenseDTO expenseDTO) | 지출보고서가 수정했을 경우 다음 결재자 emp_no 데이터를 리턴하는 메소드 선언 |
| | public int new_insertSignFinishReport_Sign_emp(ExpenseDTO expenseDTO) | 결재 후 다음결재자의 정보를 입력후 행의 개수를 리턴하는 메소드 |
| | public int alter_Next_App_Sign_Code(ExpenseDTO expenseDTO) | 다음 결재자의 사인코드를 업데이트후 적용행의 개수를 리턴하는 메소드 |

지출 보고서

| Class & Interface | Method | 기능 |
|-------------------|---|--|
| ExpenseDAOImpl | public int alter_Next_App_Sign_Date(ExpenseDTO expenseDTO) | 다음 결재자의 결재시간을 업데이트 후 적용행의 개수를 리턴하는 메소드 |
| | public String next_app_getSign_code(ExpenseDTO expenseDTO) | 다음결재자의 사인코드를 얻는 메소드 |
| | public int app_setTime(ExpenseDTO expenseDTO) | 현재결재자의 결재시간을 현재시간으로 적용 후 개수를 리턴하는 메소드 |
| | public int updateSignCodeMinus(ExpenseDTO expenseDTO) | 반려 시 사인코드를 -1로 바꾸고 적용행의 개수를 리턴하는 메소드 |
| | public int insertReturnReport(ExpenseDTO expenseDTO) | 반려시 반려사유를 입력 후 적용행의 개수를 리턴하는 메소드 |
| | public String getSign_code(ExpenseDTO expenseDTO) | 결재여부 코드 리턴 받는 메소드 |
| | public String first_app_getSign_date(ExpenseDTO expenseDTO) | 첫 결재자의 결재시간을 리턴받는 메소드 |
| | public String next_app_getSign_date(ExpenseDTO expenseDTO) | 다음 결재자의 결재시간을 리턴받는 메소드 |
| | public int alterExpenseReceipt_no(ExpenseDTO expenseDTO) | 영수증 번호 수정후 수정행의 개수를 리턴받는 메소드 |
| | public int alterExpensePrice(ExpenseDTO expenseDTO) | 지출가격 수정후 수정행의 개수를 리턴받는 메소드 |
| | public int alterRemarks(ExpenseDTO expenseDTO) | 비고란 수정후 수정행의 개수를 리턴받는 메소드 |
| | public int alterDeleteSignDate(ExpenseDTO expenseDTO) | 수정시 다음결재자의 결재시간을 삭제후 개수를 리턴받는 메소드 |
| | public int alterNextAppUpdateSignCode(ExpenseDTO expenseDTO) | 수정시 다음결재자의 사인코드를 수정후 수정행의 개수를 리턴받는 메소드 |
| | public int alterDeleteReturnReport(ExpenseDTO expenseDTO) | 수정시 반려사유를 삭제 후 삭제행의 개수를 리턴받는 메소드 |
| | public List<Map<String, String>> getAppList(ExpenseDTO expenseDTO) | 결재내역 리스트를 리턴하는 메소드 |
| | public String next_appList_getSign_date(Map<String, String> myInfo) | 결재내역에 다음 결재자의 결재시간을 리턴받는 메소드 |
| | public String my_appList_getSign_date(Map<String, String> myInfo) | 나의 결재시간을 리턴받는 메소드 |

지출 보고서

| Class & Interface | Method | 기능 |
|-------------------|--|---|
| ExpenseDTO | getter/setter 메소드로 구성 | [지출보고서]의 입력/수정/결재/반려 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |
| ExpenseSearchDTO | getter/setter 메소드로 구성 | [지출보고서]의 검색 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |
| ExpenseUtil | public static Map<String, Integer> getPagingMap(int selectPageNo, int rowCntPerPage, int totCnt) | [지출보고서]의 페이지ing 관련 데이터를 리턴하는 메소드 선언 |

관리자 페이지

| Class & Interface | Method | 기능 |
|---------------------|---|--|
| EmployeeService | EmployeeServiceImpl 클래스에서 자세히 설명 | 인터페이스로 EmployeeServiceImpl 의 메소드 목록을 규정 |
| EmployeeServiceImpl | EmployeeDTO getEmployee(int emp_no) | [직원의 상세정보]를 리턴하는 메소드 선언 |
| | int updateEmp(EmployeeDTO employeeDTO) | [직원 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언 |
| | int deleteEmp(EmployeeDTO employeeDTO) | [직원 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언 |
| | int check_emp_no(EmployeeDTO employeeDTO) | [사번 중복 체크하고 중복 적용 행의 개수]를 리턴하는 메소드 선언 |
| EmployeeDTO | getter/setter 메소드로 구성 | [직원관리 및 가입관리]의 입력/수정/삭제 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |

관리자 페이지

| Class & Interface | Method | 기능 |
|--------------------|--|--|
| EmployeeController | ModelAndView empList(EmployeeSearchDTO employeeSearchDTO, HttpServletRequest request, HttpServletResponse response, Object handler) | 가상주소 /empList.do로 접근하면 호출되는 메소드 선언 |
| | ModelAndView empDetailForm(@RequestParam(value="emp_no") int emp_no) | 가상주소 /empDetailForm.do로 접근하면 호출되는 메소드 선언 |
| | ModelAndView empRegForm() | 가상주소 /empRegForm.do로 접근하면 호출되는 메소드 선언 |
| | int empRegProc(EmployeeDTO employeeDTO), @RequestMapping(value="/empRegProc.do", method=RequestMethod.POST, produces="application/json;charset=UTF-8") json;charset=UTF-8")json;charset=UTF-8")json;charset=UTF-8") | 가상주소 /empRegProc.do로 접근하면 호출되는 메소드 선언 |
| | ModelAndView empUpDelForm(@RequestParam(value="emp_no") int emp_no) | 가상주소 /empUpDelForm.do로 접근하면 호출되는 메소드 선언 |
| | int empUpProc(EmployeeDTO employeeDTO), @RequestMapping(value="/empUpProc.do", method=RequestMethod.POST, produces="application/json;charset=UTF-8") | 가상주소 /empUpProc.do로 접근하면 호출되는 메소드 선언 |
| | int empDelProc(EmployeeDTO employeeDTO), @RequestMapping(value="/empDelProc.do", method=RequestMethod.POST, produces="application/json;charset=UTF-8") | 가상주소 /empDelProc.do로 접근하면 호출되는 메소드 선언 |
| | int emp_no_check(EmployeeDTO employeeDTO), @RequestMapping(value="/emp_no_check.do", method=RequestMethod.POST, produces="application/json;charset=UTF-8") | 가상주소 /emp_no_check.do로 접근하면 호출되는 메소드 선언 |

관리자 페이지

| Class & Interface | Method | 기능 |
|-------------------|---|--|
| EmployeeDAO | EmployeeDAOImpl 클래스에서 자세히 설명 | 인터페이스로 EmployeeDAOImpl의 메소드 목록을 규정 |
| | List<Map<String, String>> getEmpList(EmployeeSearchDTO employeeSearchDTO) | [직원 및 가입 관리 페이지 검색 리스트]를 리턴하는 메소드 선언 |
| | EmployeeDTO getEmployee(int emp_no) | [직원의 상세정보]를 리턴하는 메소드 선언 |
| | int getEmpListTotCnt(EmployeeSearchDTO employeeSearchDTO) | [검색한 사원 목록]의 총개수 리턴하는 메소드 선언 |
| | int getEmpListAllTotCnt() | [사원 목록]의 총 개수 리턴하는 메소드 선언 |
| | int insertEmp(EmployeeDTO employeeDTO) | [직원 정보 입력 후 입력 적용행의 개수]를 리턴하는 메소드 선언 |
| | getEmpCnt(int emp_no) | [수정/삭제할 직원 존재여부 개수] 리턴하는 메소드 선언 |
| | int updateEmp(EmployeeDTO employeeDTO) | [직원 정보 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언 |
| | int deleteEmp(EmployeeDTO employeeDTO) | [직원 정보 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언 |
| | int check_emp_no(EmployeeDTO employeeDTO) | [사번 중복체크 후 중복행의 적용 개수]를 리턴하는 메소드 선언 |
| EmployeeDTO | getter/setter 메소드로 구성 | [직원관리 및 가입관리]의 입력/수정/삭제 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |
| EmployeeSearchDTO | getter/setter 메소드로 구성 | [직원관리 및 가입관리]의 검색 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |

마이페이지

| Class & Interface | Method | 기능 |
|-------------------|---|--|
| MypageController | ModelAndView mypageMain(EmployeeDTO employeeDTO, HttpServletRequest request, HttpServletResponse response) | 가상주소 /mypageMain.do 로 접근하면 호출되는 메소드 선언 |
| | ModelAndView mypageUpdate(HttpServletRequest request, HttpServletResponse response, @RequestParam(value="emp_no") int emp_no) | 가상주소 /mypageUpdate.do 로 접근하면 호출되는 메소드 선언 |
| | int mypageUpProc(EmployeeDTO employeeDTO), @RequestMapping(value="/mypageUpProc.do", method=RequestMethod.POST, produces="application/json;charset=UTF-8") | 가상주소 /mypageUpProc.do 로 접근하면 호출되는 메소드 선언 |
| MypageDAO | MypageDAOImpl 클래스의 메소드와 동일 | 인터페이스로 MypageDAOImpl 의 메소드 목록을 규정 |
| MypageDAOImpl | Map<String, String> getMypage(String uid) | [로그인한 직원의 정보]를 리턴하는 메소드 선언 |
| MypageService | MypageServiceImpl 클래스의 메소드와 동일 | 인터페이스로 MypageServiceImpl 의 메소드 목록을 규정 |
| MypageServiceImpl | EmployeeDTO getMypage(int emp_no) | [로그인한 직원의 정보]를 리턴하는 메소드 선언 |

공지사항

| Class & Interface | Method | 기능 |
|-------------------|---|---|
| BoardController | ModelAndView boardList(BoardSearchDTO boardSearchDTO) | 가상주소 /boardList.do 로 접근하면 호출되는 메소드 선언 |
| | ModelAndView boardRegForm(HttpServletRequest request) | 가상주소 /boardRegForm.do 로 접근하면 호출되는 메소드 선언 |
| | Map<String, String> boardRegProc(BoardDTO boardDTO, BindingResult bindingResult), @RequestMapping(value="/boardRegProc.do", method=RequestMethod.POST, produces="application/json;charset=UTF-8") | 가상주소 /boardRegProc.do 로 접근하면 호출되는 메소드 선언 |
| | ModelAndView boardDetailForm(@RequestParam(value="b_no") int b_no) | 가상주소 /boardDetailForm.do 로 접근하면 호출되는 메소드 선언 |
| | ModelAndView boardUpDelForm(@RequestParam(value="b_no") int b_no) | 가상주소 /boardUpDelForm.do 로 접근하면 호출되는 메소드 선언 |
| | int boardUpProc(BoardDTO boardDTO), @RequestMapping(value="/boardUpProc.do", method=RequestMethod.POST, produces="application/json;charset=UTF-8") | 가상주소 /boardUpProc.do 로 접근하면 호출되는 메소드 선언 |
| | Map<String, String> boardUpProc2(BoardDTO boardDTO, BindingResult bindingResult), @RequestMapping(value="/boardUpProc2.do", method=RequestMethod.POST, produces="application/json;charset=UTF-8") | 가상주소 /boardUpProc2.do 로 접근하면 호출되는 메소드 선언 |
| BoardDAO | BoardDAOImpl 클래스에서 자세히 설명. | 인터페이스로 BoardDAOImpl의 메소드 목록을 규정 |

공지사항

| Class & Interface | Method | 기능 |
|-------------------|---|---|
| BoardDAOImpl | List<Map<String, String>> getBoardList(BoardSearchDTO boardSearchDTO) | [검색한 게시판 목록] 리턴하는 메소드 선언 |
| | int getBoardListTotCnt(BoardSearchDTO boardSearchDTO) | [검색한 게시판 목록]의 개수 리턴하는 메소드 선언 |
| | int insertBoard(BoardDTO boardDTO) | [게시판 글 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언 |
| | BoardDTO getBoard(int b_no) | [1개 게시판 글 정보] 리턴하는 메소드 선언 |
| | int getBoardCnt(int b_no) | [수정/삭제할 1개 게시판 글의 존재여부 개수] 리턴하는 메소드 선언 |
| | int updateBoard(BoardDTO boardDTO) | [1개 게시판 수정 후 수정 적용행의 개수]를 리턴하는 메소드 선언 |
| | int updateReadcount(int b_no) | [게시판 글 조회수 증가하고 수정행의 개수] 리턴하는 메소드 선언 |
| | int deleteBoard(BoardDTO boardDTO) | [1개 게시판 글 삭제 후 삭제 적용행의 개수]를 리턴하는 메소드 선언 |
| BoardDTO | getter/setter 메소드로 구성 | [공지사항 게시판]의 입력/수정/삭제 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |
| BoardSearchDTO | getter/setter 메소드로 구성 | [공지사항 게시판]의 검색 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |
| BoardService | BoardServiceImpl 클래스의 메소드와 동일 | 인터페이스로 BoardServiceImpl 의 메소드 목록을 규정 |
| BoardServiceImpl | BoardDTO getBoard(int b_no , boolean isBoardDetailForm) | [1개 게시판 글] 리턴하는 메소드 선언 |
| | int insertBoard(BoardDTO boardDTO) | [1개 게시판 글 입력 후 입력 적용 행의 개수] 리턴하는 메소드 선언 |
| | int updateBoard(BoardDTO boardDTO) | [1개 게시판] 수정 후 수정 적용행의 개수를 리턴하는 메소드 선언 |
| | int deleteBoard(BoardDTO boardDTO) | [1개 게시판] 삭제 후 삭제 적용행의 개수를 리턴하는 메소드 선언 |
| BoardValidator | void validate(Object obj, Errors errors) | 유효성 체크할 메소드 선언 |

회의실 예약

| Class & Interface | Method | 기능 |
|--------------------|--------------------|---|
| MeetingController | meetingResForm | 회의장 예약화면으로 이동하는 메소드 |
| | meetingDetailForm | 예약 게시판 상세화면으로 이동하는 메소드 |
| | meetingUpDelForm | 수정 화면으로 이동하는 메소드 |
| | meetingBoardList | 회의장 예약 게시판으로 이동하는 메소드 |
| | meetingResProc | 회의장 예약페이지에서 예약 버튼 클릭시 실행되는 메소드 |
| | meetingUpProc | 예약 수정시 실행되는 메소드 |
| | meetingDelProc | 예약 삭제시 실행되는 메소드 |
| | getCanResRoom | 예약폼에 입력양식이 모두 입력됐을경우 예약가능한 회의실 목록을 가져오는 메소드 |
| MeetingServiceImpl | insertMeetingBoard | 회의장 예약 성공시 1을 리턴하는 메소드 |
| | getTotcnt | 예약된 회의장 게시물 총 개수 리턴하는 메소드 |
| | getBoardList | 예약된 회의장 List를 가져오는 메소드 |
| | getCanResRoomList | 예약 가능한 회의장 List를 가져오는 메소드 |
| | getSearchCnt | 검색 개수를 리턴하는 메소드 |
| | updateMeetingBoard | 예약 수정 메소드 |
| | getBoard | 게시판 상세화면 정보를 가져오는 메소드 |
| | deleteMeetingBoard | 예약 삭제 메소드 |
| MeetingDAOImpl | insertMeetingBoard | 회의장 예약 성공시 1을 리턴하는 메소드 |
| | getTotcnt | 예약된 회의장 게시물 총 개수 리턴하는 메소드 |
| | getBoardList | 예약된 회의장 List를 가져오는 메소드 |
| | getCanResRoomList | 예약 가능한 회의장 List를 가져오는 메소드 |
| | getSearchCnt | 검색된 게시물 총개수를 리턴하는 메소드 |
| | getBoard | 상세화면 목록을 가져오는 메소드 |
| | updateMeetingBoard | 예약 수정하는 메소드 |
| | deleteMeetingBoard | 예약 삭제하는 메소드 |

일정 관리

| Class & Interface | Method | 기능 |
|--------------------|--------------------|---|
| ScheduleController | planListForm | 클라이언트가 가상주소 [/planList.do]에 접속 시 호출되는 메소드선언 |
| | scheduleInsertProc | 클라이언트가 가상주소 [/scheduleInsertProc.do]에 접속 시 호출되는 메소드선언 |
| | getCallInfo | 클라이언트가 가상주소 [/getCallInfo.do]에 접속 시 호출되는 메소드선언 |
| | calUpProc | 클라이언트가 가상주소 [/calUpProc.do]에 접속 시 호출되는 메소드선언 |
| | calDelProc | 클라이언트가 가상주소 [/calDelProc.do]에 접속 시 호출되는 메소드선언 |
| | calDetailPop | 클라이언트가 가상주소 [/calDetail.do]에 접속 시 호출되는 메소드선언 |
| | callInputPop | 클라이언트가 가상주소 [/callInputPop.do]에 접속 시 호출되는 메소드선언 |
| | calSharingPop | 클라이언트가 가상주소 [/calSharing.do]에 접속 시 호출되는 메소드선언 |
| | getShareOne | 클라이언트가 가상주소 [/getShareOne.do]에 접속 시 호출되는 메소드선언 |
| | getNameForm | 클라이언트가 가상주소 [/getNameForm.do]에 접속 시 호출되는 메소드선언 |
| | calShareProc | 클라이언트가 가상주소 [/calShareProc.do]에 접속 시 호출되는 메소드선언 |
| | newsDetailForm | 클라이언트가 가상주소 [/newsDetailForm.do]에 접속 시 호출되는 메소드선언 |
| | getDetailData | 클라이언트가 가상주소 [/newsDetailProc.do]에 접속 시 호출되는 메소드선언 |
| | calShareDelProc | 클라이언트가 가상주소 [/calShareDelProc.do]에 접속 시 호출되는 메소드선언 |
| | calSharePutProc | 클라이언트가 가상주소 [/calSharePutProc.do]에 접속 시 호출되는 메소드선언 |
| | calCompletProc | 클라이언트가 가상주소 [/calCompletProc.do]에 접속 시 호출되는 메소드선언 |

일정 관리

| Class & Interface | Method | 기능 |
|---------------------|-----------------------|---|
| ScheduleServiceImpl | getList | 해당 아이디의 일정목록들을 리턴하는 메소드 선언 |
| | insertList | 일정등록 데이터들을 insert 한 이후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getCalDelCnt | 일정을 delete 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getCalUpCnt | 일정등록 데이터들을 update 한 이후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getCalList | 해당 번호의 일정 목록을 리턴하는 메소드 선언 |
| | getEmpList | 해당 아이디를 제외한 모든 직원의 목록을 리턴하는 메소드 선언 |
| | getCnt | 해당 데이터를 일정공유 테이블에 insert 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getShareList | 해당 아이디로 들어온 사원의 알림 리스트를 리턴하는 메소드를 선언 |
| | getSearchCnt | 검색된 알림게시물의 개수를 리턴하는 메소드 선언 |
| | getNotifiTotCnt | 해당 사원의 알림 총 개수를 리턴하는 메소드 선언 |
| | getOneNotifiList | 해당 알림의 목록을 리턴하는 메소드 선언 |
| | getFixedAcceptanceCnt | 일정 공유 수락 시 해당 일정을 insert 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getFixedRefusalCnt | 일정 공유 거절 시 해당 알림목록을 delete 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getName | 공유한 사람의 이름을 리턴하는 메소드 선언 |
| | getCompletCnt | 완료된 일정을 update 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |

일정 관리

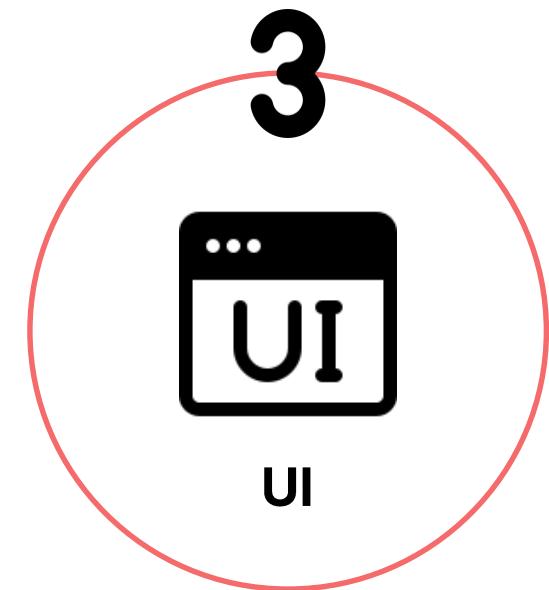
| Class & Interface | Method | 기능 |
|-------------------|-------------------------|---|
| ScheduleDAOImpl | getList | 해당 아이디의 일정목록들을 리턴하는 메소드 선언 |
| | insertList | 일정등록 데이터들을 insert 한 이후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getCalDelCnt | 일정을 delete 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getCalUpCnt | 일정등록 데이터들을 update 한 이후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getCalList | 해당 번호의 일정 목록을 리턴하는 메소드 선언 |
| | getEmpList | 해당 아이디를 제외한 모든 직원의 목록을 리턴하는 메소드 선언 |
| | getCnt | 해당 데이터를 일정공유 테이블에 insert 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getShareList | 해당 아이디로 들어온 사원의 알림 리스트를 리턴하는 메소드를 선언 |
| | getSearchCnt | 검색된 알림게시물의 개수를 리턴하는 메소드 선언 |
| | getNotifiTotCnt | 해당 사원의 알림 총 개수를 리턴하는 메소드 선언 |
| | getOneNotifiList | 해당 알림의 목록을 리턴하는 메소드 선언 |
| | getFixedAcceptanceCnt | 일정 공유 수락 시 해당 일정을 insert 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getFixedRefusalCnt | 일정 공유 거절 시 해당 알림목록을 delete 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getName | 공유한 사람의 이름을 리턴하는 메소드 선언 |
| ScheduleService | getCompletCnt | 완료된 일정을 update 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getCalShareCnt | 해당일정을 공유 했는지 유무를 리턴하는 메소드 선언 |
| | ScheduleService 클래스와 동일 | 인터페이스로 LoginServiceImpl 의 메소드 목록을 규정 |
| | ScheduleDAO 클래스와 동일 | 인터페이스로 LoginServiceImpl 의 메소드 목록을 규정 |
| ScheduleDTO | getter/setter 메소드로 구성 | 일정관리의 입력 / 수정 / 삭제 / 검색 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |
| NotificatonDTO | getter/setter 메소드로 구성 | 일정관리를 공유한 일정 알림목록을 입력 / 수정 / 삭제 / 검색 관련 데이터를 담아 이동하는 getter/setter 메소드 선언 |

회원가입

| Class & Interface | Method | 기능 |
|-------------------|-------------------------|---|
| JoinCtroller | memberJoinForm | 클라이언트가 가상주소 [/joinForm.do] 에 접속 시 호출되는 메소드선언 |
| | joinMember | 클라이언트가 가상주소 [/joinProc.do] 에 접속 시 호출되는 메소드선언 |
| | memberIdProc | 클라이언트가 가상주소 [/joinForm.do] 에 접속 시 호출되는 메소드선언 |
| | idView | 클라이언트가 가상주소 [/idView.do] 에 접속 시 호출되는 메소드선언 |
| JoinServiceImpl | getJoinCnt | 회원가입 신청할 데이터를 insert 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getId | 회원가입 후 부여된 아이디를 리턴하는 메소드 선언 |
| JoinService | JoinServiceImpl 클래스와 동일 | 인터페이스로 LoginServiceImpl 의 메소드 목록을 규정 |
| JoinDAOImpl | getJoinCnt | 회원가입 신청할 데이터를 insert 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| | getId | 회원가입 후 부여된 아이디를 리턴하는 메소드 선언 |
| JoinDAO | JoinDAOImpl 클래스와 동일 | 인터페이스로 LoginServiceImpl 의 메소드 목록을 규정 |

로그인

| Class & Interface | Method | 기능 |
|-------------------|---|---|
| LoginController | ModelAndView loginImpl | 클라이언트가 가상주소 [/loginForm.do] 에 접속 시 호출되는 메소드선언 |
| | int loginProc | 클라이언트가 가상주소 [/loginProc.do] 에 접속 시 호출되는 메소드선언 |
| | ModelAndView pwdFindForm | 클라이언트가 가상주소 [/pwdFindForm.do] 에 접속 시 호출되는 메소드선언 |
| | ModelAndView idFindForm | 클라이언트가 가상주소 [/idFindForm.do] 에 접속 시 호출되는 메소드선언 |
| | ModelAndView pwdChangeForm | 클라이언트가 가상주소 [/pwdChangeForm.do] 에 접속 시 호출되는 메소드선언 |
| | Map<String, String> findPwdProc | 클라이언트가 가상주소 [/findPwdProc] 에 접속 시 호출되는 메소드선언 |
| | Map<String, String> findIdProc | 클라이언트가 가상주소 [/findIdProc] 에 접속 시 호출되는 메소드선언 |
| | int changePwdProc | 클라이언트가 가상주소 [/changePwdProc.do] 에 접속 시 호출되는 메소드선언 |
| LoginService | LoginServiceImpl 클래스의 메소드와 동일 | 인터페이스로 LoginServiceImpl 의 메소드 목록을 규정 |
| LoginServiceImpl | int getLoginCheck | 아이디 암호의 존재여부 개수를 리턴하는 메소드 선언 |
| | String getPwd | 해당 아이디의 비밀번호를 리턴하는 메소드 선언 |
| | String getId | 해당 데이터의 아이디를 리턴하는 메소드 선언 |
| | int getChangeCnt(Map<String, String> map) | 기존의 비밀번호를 update 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |
| LoginDAO | LoginDAOImpl 클래스의 메소드와 동일 | 인터페이스로 LoginDAOImpl 의 메소드 목록을 규정 |
| LoginDAOImpl | int getLoginCheck | 아이디 암호의 존재여부 개수를 리턴하는 메소드 선언 |
| | String getPwd | 해당 아이디의 비밀번호를 리턴하는 메소드 선언 |
| | int exId | 해당 아이디의 존재여부 개수를 리턴하는 메소드 선언 |
| | String getId | 해당 데이터의 아이디를 리턴하는 메소드 선언 |
| | int exPwd | 해당 데이터를 가진 아이디의 존재여부 개수를 리턴하는 메소드 선언 |
| | int getChangeCnt | 기존의 비밀번호를 update 한 후 적용된 행의 개수를 리턴하는 메소드 선언 |



Dev Pixel

로그인



[아이디 | 비밀번호 찾기](#)

[아이디/비밀번호 저장](#)

[로그인](#) [회원가입](#)

회원가입



— 소속부서 선택 —

주민번호 앞자리 - 주민번호 뒷자리

이메일을 입력해주세요 — 도메인 선택 —

직접입력

— 휴대폰 앞자리 — - 휴대폰 번호 - 휴대폰 번호

[회원가입](#) [다시 작성](#) [로그인 화면](#)

대시보드

Homepage

한상경 대리님

공지사항

| 제목 | 등록일 |
|------------------|------------|
| [공지] 휴면보험금 서민... | 2022-12-14 |
| [공지] 휴면보험금 서민... | 2022-12-14 |
| [공지] 휴면보험금 서민... | 2022-12-14 |
| [공지] 판매보조자료 인... | 2022-12-14 |
| [공지] "코로나19" ... | 2022-12-14 |

부서별 사원 수

| 부서 | 비율 (%) |
|----|--------|
| 사무 | 30.4% |
| 행정 | 34.8% |
| 점대 | 17.4% |
| 회계 | 8.7% |
| 구매 | 2.9% |
| 기타 | 0.0% |

보고 받은 결재 처리현황

| 결재 | 1 |
|-----|---|
| 미결재 | 1 |
| 반려 | 1 |

등록한 보고서 분야별 개수 그래프

| 분야 | 개수 |
|----|----|
| 행정 | 1 |
| 사무 | 2 |
| 구매 | 0 |
| 점대 | 0 |
| 회계 | 1 |

- ① 관리자가 작성한 최근 5개 공지사항 게시
- ② 부서별 사원수 %를 나타내는 파이 차트
- ③ 현재 로그인한 사원의 보고서 종류별 결재 처리 현황
- ④ 현재 로그인한 사원의 분야별 등록보고서 개수

비밀번호 찾기

비밀번호 찾기

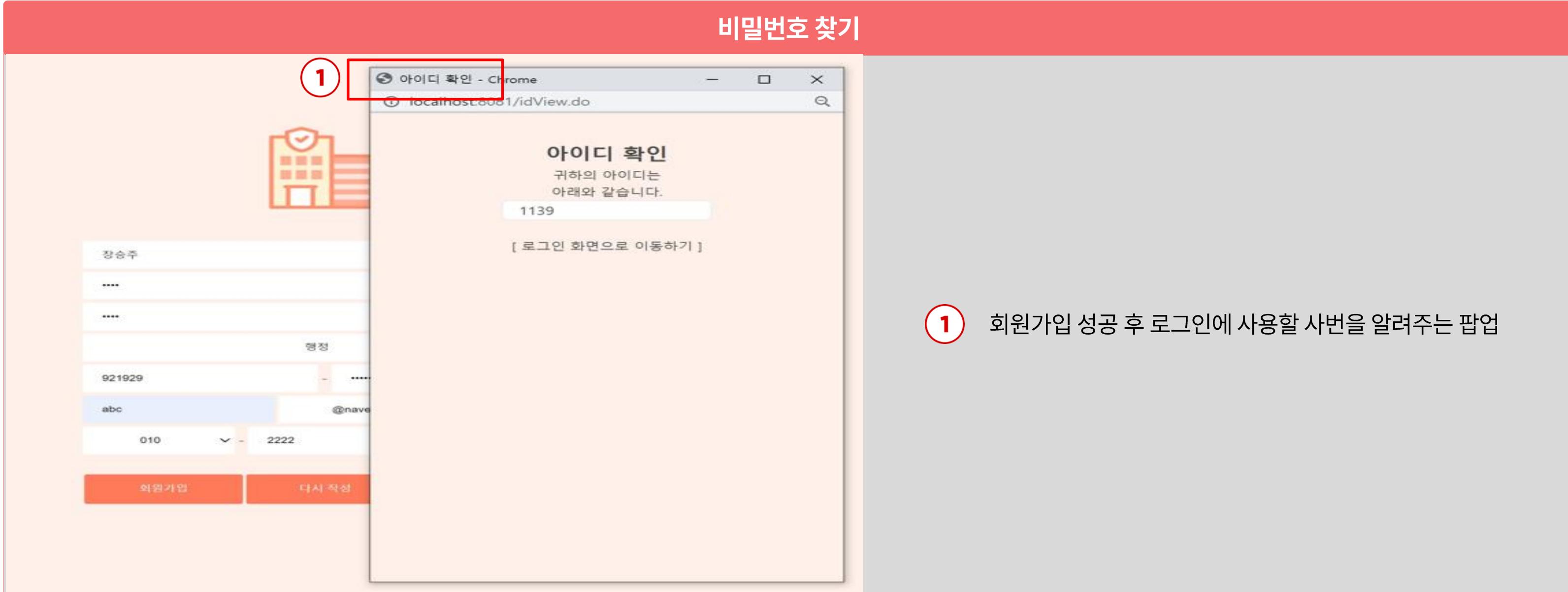
비밀번호를 찾고자하는 아이디를 입력해주세요

① 아이디 입력

② 찾기 암호변경 닫기

③ [아이디 찾기]

① 비밀번호를 찾을 사번 입력
② 일치하는 비밀번호 Return
③ 아이디 찾기 팝업창으로 이동



일일 업무 보고서 - 등록자

일일업무보고서 등록

작성자 장태하

업무분야 행정 ①

업무내용 결재 일지

시간 2022 년 12 월 13 일 03 시 01 분부터
2022 년 12 월 13 일 20 시 13 분 까지 [오늘로] [비움]

다음결재자 사무부 강건 과장 ②

등록 화면닫기 초기화

① 보고서 등록에 필요한 업무분야 선택
② 현재 보고서의 다음 결재자 선택

일일 업무 보고서 - 등록자

일일업무보고서

장태하 님

업무일 --선택-- 년 --선택-- 월 --선택-- 년 --선택-- 월 [날짜비움] [오늘날]

업무시간 -선택- 시 -선택- 분 -선택- 시 -선택- 분 [시간비움] [오늘날]

업무분야 행정 사무 구매 접대 회계 기타 [비움]

결재여부 결재완료 미결제(or 결재진행중) 반려 [비움]

키워드 [비움]

[검색] **[모두검색]** **[초기화]**

[처음] [이전] 1 [다음] [마지막] [검색 총개수] : 3/3 5 행보기

1 **업무보고서 등록**

| 번호 | 작성자 | 업무분야 | 업무내용 | 결제여부 | 결제/보류일 | 날짜 |
|----|-----|------|--------------|--------|--|--------------------|
| 3 | 장태하 | 행정 | 일일 업무 결재 테스트 | 결제 진행중 | 2022-12-13 20:11 화 ~ 2022-12-13 17:03 화 | 2022-01-01 01:00 토 |
| 2 | 장태하 | 사무 | 테스트내용 | 결재최종완료 | 2022-12-10 23:49 토 ~ 2022-12-10 23:46 토 | 2022-01-01 01:00 토 |
| 1 | 장태하 | 구매 | 테스트 | 결재최종완료 | 2022-12-10 23:21 토 ~ 2022-12-10 20:03 토 | 2022-01-01 01:00 토 |

2 업무보고서 등록 시 보고서 관련
결재라인 및 정보 안내 게시판

일일 업무 보고서 - 다음 결재자

일일업무보고서

강건 님

업무일 년 월 년 월 [날짜비움] [오늘날]

업무시간 시 분 - 시 분 [시간비움] [오늘날]

업무분야 행정 사무 구매 접대 회계 기타 [비움]

결재여부 결재완료 미결제(or 결재진행중) 반려 [비움]

키워드 [비움]

업무보고서 등록
[처음] [이전] 1 [다음] [마지막]
[검색 총개수] : 3/3
5 행보기

| 번호 | 작성자 | 업무분야 | 업무내용 | 결제여부 | 결제/보류일 | 날짜 |
|----|-----|------|--------------|--------|--|----|
| 1 | 장태하 | 행정 | 일일 업무 결재 테스트 | 미결재 | 2022-01-01 01:00 토 ~ 2022-12-13 17:03 화 | |
| 2 | 장태하 | 사무 | 테스트내용 | 결재최종완료 | 2022-12-10 23:49 토 ~ 2022-12-10 23:46 토 | |
| 3 | 장태하 | 구매 | 테스트 | 결재완료 | 2022-12-10 23:21 토 ~ 2022-12-10 20:03 토 | |

① 업무보고서 미결재 상태

② 업무보고서 최종 결재가 완료된 상태

③ 현재 로그인한 ID에서 결재한 상태

일일 업무 보고서 - 다음 결재자

일일업무보고서

작성자: 장태하 강건 님

업무 분야: 행정

업무 내용: 일일 업무 결재 테스트

시간: 2022년 01월 01일 01시 00분 부터
 2022년 12월 13일 17시 03분 까지

결재/반려: ○ 결재 ○ 반려

1 다음 결재자 선택창

2 현재 결재라인 현황

다음 결재자: 장태하 다음결재자 없음

| 결재 내역 | 결재순서 | 결재자명 | 부서명 | 직책 | 결재여부 | 결재일 | 반려사유 |
|-------|------|------|-----|-----|--------------------|-----|------|
| 1 | 장태하 | 행정 | 대리 | 결재 | 2022-12-13 20:11 화 | | |
| 2 | 강건 | 사무 | 과장 | 미결재 | | | |

결재/반려 화면닫기

일일 업무 보고서 - 등록자

일일업무보고서

| 업무일 | --선택-- 년 | --선택-- 월 | --선택-- 년 | --선택-- 월 [날짜비율] [오늘날] | 장태하 님 |
|---|--|--|--------------|---|--|
| 업무시간 | -선택- 시 | -선택- 분 | -선택- 시 | -선택- 분 [시간비율] [오늘날] | |
| 업무분야 | <input type="checkbox"/> 행정 <input type="checkbox"/> 사무 <input type="checkbox"/> 구매 <input type="checkbox"/> 접대 <input type="checkbox"/> 회계 <input type="checkbox"/> 기타 [비음] | | | | |
| 결재여부 | <input type="checkbox"/> 결재완료 <input type="checkbox"/> 미결제(or 결재진행중) <input type="checkbox"/> 반려 [비음] | | | | |
| 키워드 | <input type="text" value=""/> [비율] | | | | |
| 검색 모두검색 초기화 | | | | | |
| 업무보고서 목록 | | [처음] [이전] 1 [다음] [마지막] | | [검색 총개수] : 3/3 5 ▼ 행보기 | |
| 번호 | 작성자 | 업무분야 | 업무내용 | 결재여부 | 결제/보류일 |
| 3 | 장태하 | 행정 | 일일 업무 결재 테스트 | 결재최종완료 | 2022-12-13 20:20 화 ~ 2022-12-13 17:03 화 |
| 2 | 장태하 | 사무 | 테스트내용 | 결재최종완료 | 2022-12-10 23:49 토 ~ 2022-12-10 23:46 토 |
| 1 | 장태하 | 구매 | 테스트 | 결재최종완료 | 2022-12-10 23:21 토 ~ 2022-12-10 20:03 토 |

① 결재 최종 완료 후 결재여부 상태

①

지출 보고서 – 등록자

지출보고서등록

장태하 님

| 지출 제목 | 분야 | 수단 | 비용 | 영수증번호 |
|--------|----|----|----|-------|
| | | | | 원 |
| 지출일 | | | 비고 | |
| | 년 | 월 | 일 | |
| 다음 결재자 | | | | |

등록 **화면닫기** **초기화**

지출 보고서 – 등록자

지출보고서
장태하 님

지출일 년 월 년 월 [날짜비움] [오늘날]

결재여부 결재완료 미결제(or 결재진행중) 반려 [비움]

지출수단 회사카드 회사현금 개인카드 개인현금 [비움]

지출분야 식대 회식비 접대비 비품 기자재 사무기구 교통비 경조사비 숙박비 의류비 도서비 홍보비 기타 [비움]

비용 원 ~ 원 [비움]

키워드 [비움]

[처음] [이전] 1 [다음] [마지막]
[검색 총개수] : 1/1
5 행보기

| 번호 | 작성자 | 지출제목 | 지출분야 | 결재여부 | 결재/반려일 | 지출일 | 비용 | 지출수단 |
|----|-----|------|------|--------|--------------------|------------|--------|------|
| 1 | 장태하 | 테스트 | 회식비 | 결제 진행중 | 2022-12-11 00:43 일 | 2022-12-07 | 120000 | 회사카드 |

① 지출 보고서 등록 후 결재여부 상태

영업 보고서 – 등록자

영업 보고서 등록

장태하 님

제목

방문 목적

만난 사람

| 회사이름 | 이름 | 직급 | 전화번호 |
|----------------------|----------------------|----------------------|----------------------|
| <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |

방문 시간

년 월 일 시 분 부터

년 월 일 시 분 까지 [오늘로] [비움]

영업 내용

영업 성과

다음 결제자

등록

화면 닫기

초기화

영업 보고서 – 다음 결재자

영업보고서

영업일 --선택-- 년 --선택-- 월 - --선택-- 년 --선택-- 월 [날짜비움] [오늘날]

영업시간 -선택- 시 -선택- 분 - -선택- 시 -선택- 분 [시간비움] [오늘날]

결재여부 결재완료 미결제(or 결재진행중) 반려 [비움]

방문목적 영업자관리 수급 회의 거래처확보 기타 [비움]

키워드 [비움]

검색 모두검색 초기화

업무보고서 등록

[처음] [이전] 1 [다음] [마지막] [검색 총개수] : 1/1 5 행보기

| 번호 | 작성자 | 영업제목 | 방문목적 | 결재여부 | 결재/반려일 | 영업일 |
|----|-----|------|------|------|--------------------|----------------------|
| 1 | 장태하 | 테스트 | 회의 | 미결재 | 2022-12-06 06:03 화 | ~ 2022-12-06 15:05 화 |

① 영업 보고서 등록 후 결재여부 상태

일정관리

일정관리 장태하 님

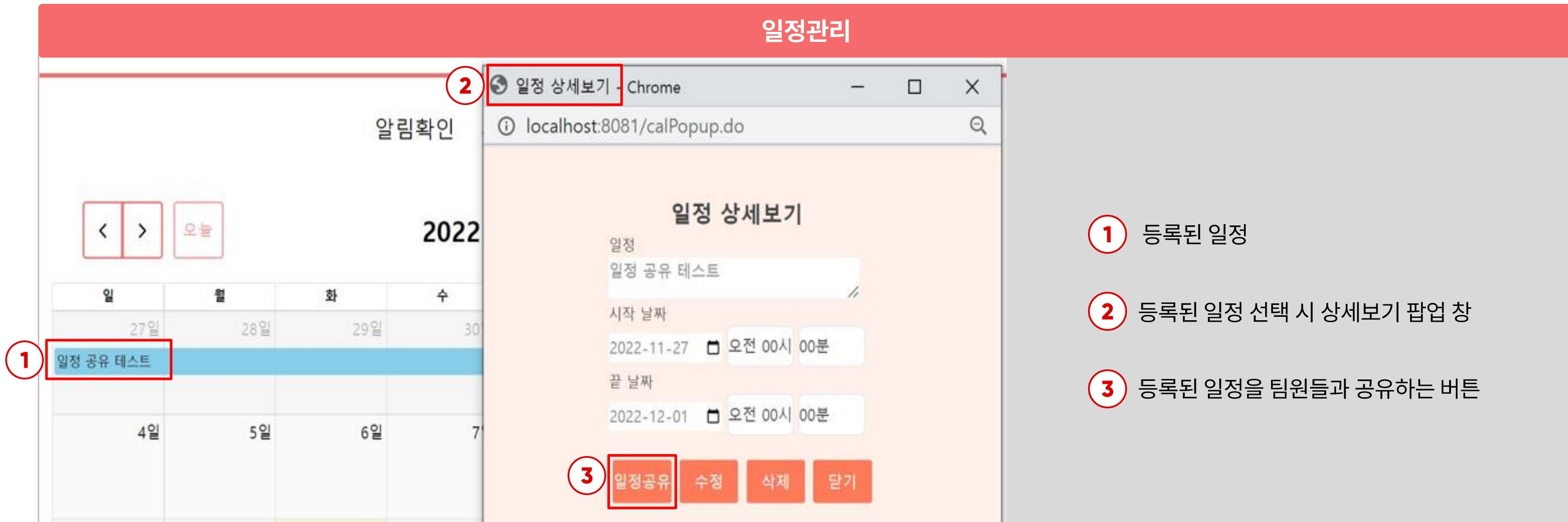
알림확인 0

< > 오늘 월

2022년 12월

| 일 | 월 | 화 | 수 | 목 | 금 | 토 |
|-----|-----|-----|-----|-----|-----|-----|
| 27일 | 28일 | 29일 | 30일 | 1일 | 2일 | 3일 |
| 4일 | 5일 | 6일 | 7일 | 8일 | 9일 | 10일 |
| 11일 | 12일 | 13일 | 14일 | 15일 | 16일 | 17일 |
| 18일 | 19일 | 20일 | 21일 | 22일 | 23일 | 24일 |
| 25일 | 26일 | 27일 | 28일 | 29일 | 30일 | 31일 |
| 1일 | 2일 | 3일 | 4일 | 5일 | 6일 | 7일 |

① 현재 날짜
② 일정 등록 시 파란색 Bar 표기





일정공유

일정관리

1 강건 님

알림확인 2

< > 오늘

2022년 12월 월

| 일 | 월 | 화 | 수 | 목 | 금 | 토 |
|-----|-----|-----|-----|-----|-----|-----|
| 27일 | 28일 | 29일 | 30일 | 1일 | 2일 | 3일 |
| 4일 | 5일 | 6일 | 7일 | 8일 | 9일 | 10일 |
| 11일 | 12일 | 13일 | 14일 | 15일 | 16일 | 17일 |
| 18일 | 19일 | 20일 | 21일 | 22일 | 23일 | 24일 |
| 25일 | 26일 | 27일 | 28일 | 29일 | 30일 | 31일 |
| 1일 | 2일 | 3일 | 4일 | 5일 | 6일 | 7일 |

1 일정을 공유 받은 팀원으로 로그인

2 일정 공유 신청을 받을 경우 또는 알림

일정공유

일정관리

1 강건 님

알림확인 2

< > 오늘

2022년 12월 월

| 일 | 월 | 화 | 수 | 목 | 금 | 토 |
|-----|-----|-----|-----|----|----|----|
| 27일 | 28일 | 29일 | 30일 | 1일 | 2일 | 3일 |
| 7일 | 8일 | 9일 | 10일 | | | |
| 14일 | 15일 | 16일 | 17일 | | | |
| 21일 | 22일 | 23일 | 24일 | | | |
| 28일 | 29일 | 30일 | 31일 | | | |
| 4일 | 5일 | 6일 | 7일 | | | |

▲▲▲ 알람 접기 ▲▲▲

[키워드] : 검색 초기화

번호 보낸사 메시지 보낸시각

1 장태 일정 공유... 2022-12-13 (화) 오후 08:24:42

◀ ◀ 1 ▶ ▶

3

1 일정을 공유 받은 팀원으로 로그인

2 일정 공유 신청을 받을 경우 또는 알림

3 알림 창을 누를 경우 알림 확인 팝업 창

일정공유

알림확인 🔔 0

< > 오늘 월

2022년 12월

| 일 | 월 | 화 | 수 | 목 | 금 | 토 |
|-----------|-----|-----|-----|-----|-----|-----|
| 27일 | 28일 | 29일 | 30일 | 1일 | 2일 | 3일 |
| 일정 공유 테스트 | | | | | | |
| 4일 | 5일 | 6일 | 7일 | 8일 | 9일 | 10일 |
| 11일 | 12일 | 13일 | 14일 | 15일 | 16일 | 17일 |
| 18일 | 19일 | 20일 | 21일 | 22일 | 23일 | 24일 |
| 25일 | 26일 | 27일 | 28일 | 29일 | 30일 | 31일 |
| 1일 | 2일 | 3일 | 4일 | 5일 | 6일 | 7일 |

1 일정 공유를 수락할 경우 일정이 등록되고
공유 받은 일정은 붉은색 Bar 표기

회의실 예약

회의실 예약

강건 님

예약목록

 예약자,제목으로 검색 소속부서 직급 검색 모두검색 [검색 총개수] : 32/32개 10 행보기

| 번호 | 소속부서 | 예약자 | 회의제목 | 내용 | 시작 시간 | 종료 시간 |
|----|------|--------|-------|---------|-----------------------|-----------------------|
| 31 | 행정 | 대리 장태하 | 테스트 | 테스트 | 2022-12-10 09:00:00.0 | 2022-12-10 11:00:00.0 |
| 30 | 사무 | 과장 강건 | 테스트 | 테스트 | 2022-12-10 09:00:00.0 | 2022-12-10 11:00:00.0 |
| 29 | 행정 | 대리 장태하 | 테스트 | 테스트 | 2022-12-08 13:00:00.0 | 2022-12-08 15:00:00.0 |
| 28 | 행정 | 대리 장태하 | 테스트 | 1111 | 2022-12-08 09:00:00.0 | 2022-12-08 11:00:00.0 |
| 27 | 행정 | 대리 장태하 | 테스트 | 테스트 | 2022-12-07 09:00:00.0 | 2022-12-07 11:00:00.0 |
| 26 | 행정 | 사장 전소연 | 테스트 | 테스트 | 2022-12-07 09:00:00.0 | 2022-12-07 11:00:00.0 |
| 25 | 행정 | 대리 장태하 | 테스트 | 테스트 | 2022-12-06 09:00:00.0 | 2022-12-06 11:00:00.0 |
| 24 | 행정 | 대리 장태하 | 테스트 | 테스트내용 | 2022-12-04 09:00:00.0 | 2022-12-04 11:00:00.0 |
| 23 | 행정 | 대리 장태하 | 데이터확인 | 데이터확인 | 2022-01-05 01:00:00.0 | 2022-01-05 13:00:00.0 |
| 22 | 행정 | 대리 장태하 | 테스트23 | 테스트내용23 | 2022-12-01 13:00:00.0 | 2022-12-01 15:00:00.0 |

[처음] [이전] 1 [2] [3] [4] [다음] [마지막]

회의실 예약 페이지 이동

회의실 예약

회의실 예약

예약자 강건

회의제목

회의내용

① 이용시간

2022 년 12 월 13 일 시 부터
2022 년 12 월 13 일 시 까지

② 예약 가능 회의실

회의실 조회

예약 취소

강건 님

① 예약 페이지 접속 시 현재 날짜 설정
시작 시간 입력 시 종료 시간은 2시간 뒤로 설정

② 조회 버튼을 통한 예약 가능한 회의실 조회

관리자 페이지

관리자 페이지 관리자 님

가입관리

검색
모두 검색
등록
[검색 총개수] : 41개
10 행보기

| 번호 | 사번 | 이름 | 직급 | 부서 | 직속상관사번 | 직속상관이름 | 전화번호 | 가입상태 |
|----|-------|-----|----|----|--------|--------|---------------|------|
| 41 | 1139 | 장승주 | 사원 | 행정 | 0 | 없음 | 010-2222-2222 | 승인대기 |
| 40 | 1138 | 윤윤윤 | 사원 | 사무 | 0 | 없음 | 010-2222-2222 | 승인대기 |
| 39 | 1137 | 활우준 | 사원 | 행정 | 0 | 없음 | 010-2222-2222 | 가입승인 |
| 38 | 1136 | 정윤윤 | 사원 | 행정 | 0 | 없음 | 010-2222-2222 | 가입승인 |
| 37 | 1135 | 정정정 | 사원 | 행정 | 0 | 없음 | 010-2222-2222 | 가입승인 |
| 36 | 1134 | 장윤주 | 사원 | 행정 | 0 | 없음 | 010-2222-2222 | 승인대기 |
| 35 | 1133 | 한상경 | 사원 | 접대 | 1000 | 손홍민 | 010-2222-2222 | 승인대기 |
| 34 | 1132 | 정찬성 | 사원 | 사무 | 1000 | 손홍민 | 010-2222-2222 | 가입승인 |
| 33 | 10002 | 정우영 | 사원 | 행정 | 1000 | 손홍민 | 010-2222-2222 | 가입승인 |
| 32 | 10001 | 정태후 | 사원 | 행정 | 1000 | 손홍민 | 010-2222-2222 | 가입승인 |

[처음] [이전] 1 [2] [3] [4] [5] [다음] [마지막]

2

1

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

관리자 페이지 – 직원 정보 등록

직원 정보 등록

관리자 님

| | |
|----------------------|----------------------|
| 이름 | 사번 |
| <input type="text"/> | <input type="text"/> |
| 암호 | 암호확인 |
| <input type="text"/> | <input type="text"/> |
| 주민등록번호 | 이메일 |
| <input type="text"/> | <input type="text"/> |
| 권한설정 | 소속부서 |
| 일반계정 | 행정 |
| 직급 | 직속상관 |
| 사장 | <input type="text"/> |
| 전화번호 | 가입상태 |
| <input type="text"/> | 가입승인 |

등록하기 **취소**

관리자 페이지 – 직원 정보 수정

직원 정보 수정 관리자 님

| | |
|------------------------------------|------------------------------------|
| 이름 | 변경할 사번 |
| 장승주 | 1139 <small>기존 사번: 1139</small> |
| 새로운 암호 | 새로운 암호확인 |
| 1111 <small>기존 암호: 1111</small> | |
| 주민등록번호 | 이메일 |
| 9219291021211 | abc@naver.com |
| 권한설정 | 소속부서 |
| 일반계정 | 행정 |
| 직급 | 직속상관 |
| 사원 | 사장 손흥민 |
| 전화번호 | 가입상태 |
| 01022222222 | 승인대기 |

① 소속부서 변경 시 변경된 부서의 현재 직원보다 상급자 목록 출력
② 승인대기를 관리자가 가입승인으로 변경해주어야 회원가입한 사번 이용 가능

수정 취소

마이페이지

마이페이지

장태하 님



이름

장태하

사번

1021

부서

행정

직급

대리

직속상관이름

전소연

생년월일

2076년07월21일

전화번호

01011111131

이메일

abc@naver.com

수정

마이페이지

마이페이지

장태하 님

1 profile

| | |
|----------|---------------------|
| 이름 | 장태하 |
| 사번 | 1021 |
| 생년월일 | 2076년07월21일 |
| 전화번호 | 01011111131 |
| 이메일 | abc@naver.com |
| 새로운 암호 | **** 기존 암호: 1111 |
| 새로운 암호확인 | |

1 profile 클릭 시 사진 업로드 가능

수정하기 **수정취소**

4



후기

우선 모든 팀원들에게 정말 감사하다는 말을 하고 싶습니다.

모두가 각자 맡은 부분들을 기간 내에 완성해주고, 각자 파트가 끝나고도 부족한 파트를 서로 도와가며 프로젝트에 신경 써줘서 우려되었던 일들 없이 원만하게 마무리 할 수 있었던 것 같습니다.

그리고 처음 파트를 나눌 때 sql, java, javascript 등 기술 카테고리로 파트를 나누지 않고 한 파트를 처음부터 마무리까지 각자 해볼 수 있도록 나눴던게 도움이 많이 되었던 것 같습니다.

각자 맡은 파트에서 배웠던 기술들을 적용 해봄으로써 배운 내용들에 익숙해 질 수 있었고, 서로 다른 파트에서 썼던 기술들을 맡은 파트에 적용해볼 수도 있고 모르는 부분을 물어봤을 때 같은 문제를 이미 다른 사람이 경험 해본 경우도 있었기 때문에 서로 답변해주기도 수월 해질 수 있었습니다.

특히 각자 파트를 진행하게 되면서 DB가 통일 되어 있지 않아서 모든 소스를 합칠 때 입력양식의 전달 데이터에서 에러가 많이 생겼었는데, 이를 통해 협업과, 프로젝트 구상의 중요성을 다시 한번 깨달을 수 있었고 차근차근 데이터를 확인해 나가며 디버깅을 했던 경험은 앞으로 개발자로 일할 때에 밑거름이 되는 귀중한 경험 이었습니다.



정민규/리더

팀원분들과 각자 카테고리를 분담하고, 하나의 시스템을 구축하는 과정을 통해 수업시간에 배운 내용들을 다시 한번 실전에서 DB연동이 어떤 식으로 되어있는지 알 수 있었습니다.

특히 데이터베이스를 시스템에 연동할 때는 자칫하면 오류가 나서 화면이 아예 보이지 않을 때가 많아 어려움을 많이 겪었습니다.

그 때마다 서로의 상황을 공유하여 해결방안을 찾는 과정을 통해 어려움을 헤쳐 나갈 수 있었습니다.

이러한 오류 해결을 통해 팀원들과 협동심을 기를 수 있었고 해결 능력도 키우게 된 것 같습니다.

또한 DB연동을 위한 다양한 쿼리문을 작성하면서 올바른 쿼리문을 작성하는 방법을 많이 배울 수 있어서 뜻깊은 시간이었습니다.

한달이라는 기간 안에 각자 맡은 바를 성실히 임해주신 팀원분들에게 감사드립니다.



손혜인/리더



최철민/팀원

우선 함께해준 팀원들에게 너무 수고했다는 말을 해주고 싶습니다.

많은 의견을 냈음에도 너그럽게 이해해주고 수용해준 리더들에게 고맙고 같은 전자결재 파트를 맡았던 진성이와 예찬이 형에게 파트를 맡겼을 때

별탈 없이 맡은 작업을 항상 완벽하게 완수해줘서 모두가 만족한 결과물이 나올 수 있었던 것 같습니다.

또 프로젝트를 진행하면서 막히는 부분이 생길 때마다 풀어내면서 쾌감을 느꼈으며 뭐든지 할 수 있다는 자신감이 생겼고 이 자신감을 바탕으로 모든 문제를 잘 풀 수 있는 능력을 얻게 된 것 같습니다.



김진성/팀원

프로젝트를 하면서 혼자 이론으로 얻을 수 있는 학습이 아닌 팀원들을 통해 얻은 업무수행 방향과 커뮤니케이션을 많이 습득할 수 있었습니다. 팀원들 모두 프로젝트라는 큰 산이 처음 겪어봤을 텐데 많은 시행착오가 있음에도 항상 불평하지 않고 포기하지 않는 모습으로 진행해 주셔서 감사하고 행운이라 생각합니다.

프로젝트를 진행하면서 다양한 원인의 에러를 하나씩 해결해 가면서 에러에 대한 두려움이 점점 자신감으로 바뀌는 제 모습이 뿌듯했고, 가장 큰 성과라고 생각합니다.

그리고 리더분들이 앞에서 끌어주고 뒤에서 함께해 주셔서 완벽하게 성공적으로 마무리할 수 있었습니다.

모두 이번 프로젝트로 시작해서 더욱 발전된 개발자가 되었으면 좋겠습니다.



임예찬/팀원

이번 프로젝트를 통해 팀원들과 하나의 사이트를 구축했고 학원에서 배운 것들을 하나하나 적용해 보면서 다시 한번 공부할 수 있었고 프로젝트를 하며 기능을 하나씩 구현해 나가며 성취감을 느꼈습니다. 오류나 예상치 못한 일들이 생길 때는 쉽지 않았지만 더욱 성장 할 수 있는 시간이 됐습니다.

Java와 오라클이 어떠한 방식으로 연동되는지, 각각의 클래스와 인터페이스들 Controller, Service, DAO, DTO, XML, Mapper, Jsp 들이 어떠한 연관 관계를 가지고 있는지 어떠한 작업을 하기위해서는 무엇이 선행되어야 하는지 좀더 지식이 하나로 뭉쳐지고 명확해졌습니다.

학원을 다니는 기간 중에 이 프로젝트에서 가장 많은 것들을 얻을 수 있었고 개발자로 조금씩 다가갈 수 있는 유익한 시간이었습니다.

마지막으로 항상 열심히 해준 팀원들 덕분에 프로젝트를 무사히 일정 안에 끝내서 너무 고생했고 감사하다는 말을 전하고 싶습니다.



조수아/팀원

길다면 길고 짧다면 짧은 3주동안 정말 많은 것을 배운 것 같습니다. 우선, 개발자로서 나아갈 초석을 같이 다져주신 팀원 모두에게 감사하다는 말씀을 전하고 싶습니다.
이론으로써 코딩을 접했을 때는 막연하게만 느껴졌던 것들이 직접 코드를 작성하고 실행하며 제대로 이해할 수 있었고, 그 과정에서 만난 수많은 에러들 덕분에 많이 성장할 수 있었습니다.
제가 맡았던 마이페이지를 만들 때 HttpSession 객체에서 정보를 불러와 수정하는 데 어려움을 겪었습니다.
한참을 씨름하다 팀원에게 도움을 요청했고, 팀원과 여러 방법을 시도해보며 성공적으로 페이지를 만들 수 있게 되었습니다.
저는 이 과정에서 의견교환과 협업의 중요성에 대해 깨닫게 되었고, 나아가 앞으로 개발자로 살아가는데 있어 커다란 밑거름이 될 소중한 시간이었다고 생각합니다.



한상경/팀원

팀원들과 원활한 소통과 분위기 덕분에 각자 아이디어를 주고 받아 프로젝트 일정과 역할을 정하는데 차질없이 진행되어 좋은 협응력과 배려심을 가진 뛰어난 팀원들과 함께 하게 되어 너무 좋았습니다.
각자 파트를 나누어 일정을 시작하는데 생각보다 큰 압박감과 책임감을 느끼게 되어 열의가 더욱 불타올랐습니다.
제가 맡은 파트 중에서 일정관리 페이지를 구현하기 위해서 구글 달력 API를 활용하였는데 이러한 작업을 처음 해봐서 시작부터 막히면서 큰 좌절감에 휩싸이며 자존감이 많이 떨어졌습니다.
하지만 팀원들도 이러한 환경속에서도 나아가고 있다고 생각하니 더욱더 분발하여 밤을 새더라도 기능을 구현하기 위해서 계속 구글링으로 찾아보고 테스트 해보며 결국엔 달력 API를 가져와 DB와 연동하는 것부터 시작하여 달력 API의 기능들을 활용하여 원하는 입력/수정/삭제 기능을 구현하였습니다.
이러한 경험 덕분에 저희가 만드는 프로젝트에 메인 화면인 대시보드 화면에서 구글차트 API를 끌어와 DB 연동을 하여 원하는 통계를 보이게 하는데 큰 어려움 없이 구현하게 되었습니다.
프로젝트를 진행하며 가장 크게 느낀 점은 하루하루 제가 작업한 일지를 작성하는게 정말 중요하다고 생각했습니다.
아무리 밤새워 노력했다 하더라도 하루하루 지나면서 제가 어떤 문제에 막혔었고 어떠한 기능을 구현하였는지 까먹기 때문에 현업에 가서 일을 하게 되면 하루하루 일지대장을 만들어 기록해야겠다는 생각을 했고, 프로젝트를 통하여 커다란 압박감을 이겨내고 나서 얻는 이 성취감과 자신감으로 앞으로도 더욱 정진 해야겠다고 느끼게 되었습니다.



Dev Pixel