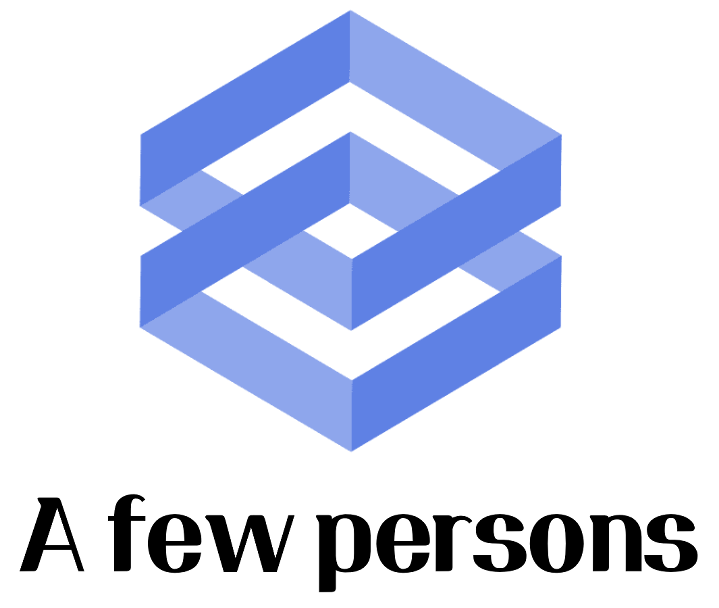
**AFPS 프로젝트 완료 보고서**



**A팀**

팀장 : 윤수진

팀원 : 김은중, 이가람, 김륜광

**목 차**

**1. 프로젝트 개요**

1-1. 프로젝트 명칭 및 기간 ·····································································································３

1-2. 프로젝트 목적 ······················································································································３

1-3. 팀원 담당 및 업무 ··············································································································４

**2. 개발환경**　········································································································································· ５

**3. 설계사양서**

3-1. ERD ··········································································································································· 6

3-2. 테이블 구조 ·····························································································································7

**·** afps\_member테이블

**·** annual\_leave테이블

**·** approval\_line 테이블

**·** board테이블

**·** commute\_management테이블

**·** data\_library테이블

**·** electronic\_approval테이블

**·** file\_info테이블

**·** manager\_email테이블

**·** reply테이블

**·** schedule\_management테이블

3-3. 디렉토리 구조 ··················································································································· 11

3-4. 패키지 구조 ························································································································ 13

**4. 어플리케이션**

4-1. 주요 기능 및 정리 ·········································································································· 14

i. 게시글 파일 다중 업로드 ······························································································· 14

ii. 부서별 사원 리스트 출력 ··························································································· １８

iii. 퇴근 등록 ··························································································································· 2０

iiii. 결재 최종승인시 캘린더 자동등록 ········································································ ２５

**5. 프로젝트 후기** ·························································································································· ３２

**1. 프로젝트 개요**

1-1. 프로젝트 명칭 및 기간

프로젝트 명칭 : A few persons

프로젝트 기간 : 2020. 05. 24 ~ 2020. 07. 27

1-2. 프로젝트 목적

기존 시장에서 사용되는 그룹웨어의 다양한 기능으로 인해 배우고 적응하는 시간이 오래 걸린다는 점을 개선하여, 부가 기능을 덜어내고 필수 기능만을 중점적으로 구현하며 명확하고 깔끔한 사용자 환경을 제공해 누구나 쉽고 편리하게 사용할 수 있는 소규모 그룹웨어

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 단계 | 업무 | | 작업자 | 작업일정 | | |
| 시작일 | 종료일 | 일수 |
| 착수 | 프로젝트 팀 구성 | 프로젝트 TFT 선정 | 전체 | 2021-05-10 | 2021-05-14 | 5 |
| 프로젝트 주제 선정 | 요구사항 도출/분석 | 전체 | 2021-05-17 | 2021-05-21 | 5 |
| 프로젝트 수행 계획 | 산출물 목록 정의 | 팀장 | 2021-05-24 | 2021-05-25 | 2 |
|  | 작업 분담/수행계획서 작성 | 전체 | 2021-05-24 | 2021-05-25 | 2 |
| 프로젝트 런칭 | 착수보고 | 팀장 | 2021-05-24 | 2021-05-25 | 2 |
| 분석 | 사이트 계획 수립 | 프로젝트 일정계획 수립 | 전체 | 2021-05-24 | 2021-05-25 | 2 |
| 설계 |  | 사이트맵 | 전체 | 2021-05-24 | 2021-05-25 | 2 |
|  | 화면구조UI / 스토리보드 설계 | 전체 | 2021-05-25 | 2021-05-28 | 4 |
|  | DB 설계 | 전체 | 2021-05-31 | 2021-06-04 | 5 |
|  | 프로그램 설계 | 전체 | 2021-06-07 | 2021-06-11 | 5 |
| 중간보고 | 프로젝트 진행상황 중간보고 | 팀장 | 2021-06-14 | 2021-06-14 | 1 |
| 구현 | 메뉴별 디자인 | 화면별 디자인 | 각 팀원 | 2021-06-21 | 2021-06-25 |  |
| 프로그램 | 공용 Lib/모듈 개발 | 전체 | 2021-06-28 | 2021-07-16 |  |
|  | 로그인 구현 | 이가람 |  |
|  | 마이 페이지 구현 | 이가람 |  |
|  | 나의 정보 구현 | 김은중 |  |
|  | 메인 페이지 구현 | 윤수진 |  |
|  | 일정관리 구현 | 김륜광 |  |
|  | 근태관리 구현 | 이가람 |  |
|  | 게시판 구현 | 윤수진 |  |
|  | 전자결재 구현 | 김륜광 |  |
|  | 자료실 구현 | 윤수진 |  |
|  | 사원정보 구현 | 김은중 |  |
|  | 관리자 구현 | 김은중 |  |
| 테스트 | 테스트 계획 | 체크리스트 작성 | 전체 | 2021-07-19 | 2021-07-20 |  |
| 테스트 수행 | 개발자 단위 테스트 | 전체 | 2021-07-19 | 2021-07-20 |  |
| 오류 수정 |  | 오류 수정 | 전체 | 2021-07-21 | 2021-07-21 |  |
| 완료 |  | 프로젝트 완료보고서 작성 | 팀장 | 2021-07-22 | 2021-07-23 |  |

1-3. 팀원 담당 및 업무

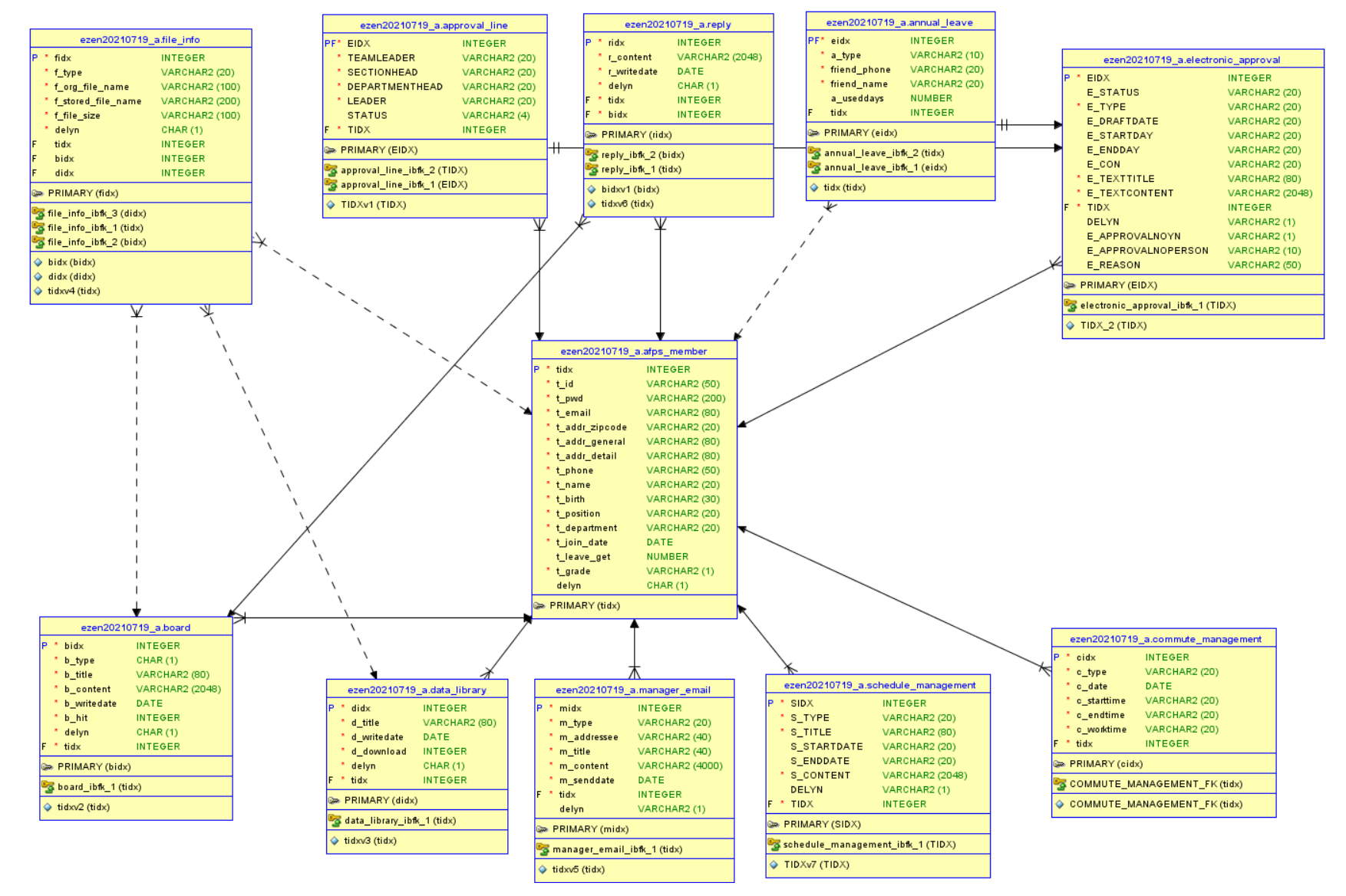
|  |  |  |
| --- | --- | --- |
| 이름 | 담당업무 | 업무 내용 |
| 윤수진 | DB 설계 및 관리 | 데이터베이스 설계 및 관리 |
| 문서 관리 | 전체 프로젝트 문서(프로젝트 완료보고서) 작성 및 관리 |
| CSS 공통모듈 | Include 용 JSP 파일 및 CSS |
| 메인페이지 | 메인 페이지 |
| 통합게시판 | 게시판 글 목록, 작성, 수정, 삭제 로직 구현 |
| 자료실 | 자료실 파일 목록, 업로드, 다운로드 로직 구현 |
| 김은중 | DB 설계 및 관리 | 데이터베이스 설계 및 관리 |
| 깃허브관리 | 깃허브 저장소 관리 |
| 페이지접근 | 페이지 접근 권한 및 관리자 권한 부여 |
| 관리자 | 사원 및 메일 조회, 등록, 수정, 삭제(퇴사) |
| 관리자 등록 로직 구현 |
| 연차관리 | 연차 등록, 수정, 삭제 로직 구현 |
| 나의정보 | 활동 기록(게시판, 일정, 연차, 메일) 조회, 등록, 수정, 삭제 로직 구현 |
| 사원정보 | 사원 목록 조회 |
| 로그아웃 | 세션값 삭제 |
| 이가람 | DB 설계 및 관리 | 데이터베이스 설계 및 관리 |
| 스토리보드 | 스토리보드 작성 및 관리 |
| 로그인 | 사원 로그인 로직 구현 |
| 비밀번호 찾기 | 임시 비밀번호 메일 전송 로직 구현 |
| 마이페이지 | 사원 정보 변경 로직 구현 |
| 비밀번호 변경 | 비밀번호 변경 로직 구현 |
| 근태관리 | 출퇴근 목록 조회 및 등록 로직 구현 |
| 김륜광 | DB 설계 및 관리 | 데이터베이스 설계 및 관리 |
| 페이지접근 | 페이지 접근 권한 및 관리자 권한 부여 |
| 일정관리 | 일정 조회, 등록, 수정, 삭제 기능 구현 |
| 전자결재 | 전자결재 시스템 로직 구현  및 일정 자동 등록 로직 구현 |
| 결재라인 | 직급별 결재권한 부여 기능 구현 |

**2. 개발환경**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 구분 | | 개발 환경 | | 서비스 환경 |
| S/W | OS | Windows 10 | | Windows 10 |
| 웹브라우저 | Chrome | | Chrome |
| WAS | Apache-tomcat-8.5 | | Apache-tomcat-8.5 |
| 개발언어 | 서버 | Java 1.8 | JDK 1.8.0  JSP 2.1 |
| 클라이언트 | javaScript  HTML5  CSS3  jsp |
| Framework | Spring | | - |
| DBMS | MySQL | | - |
| Tools | Maven | | - |

**3. 설계사양서**

3-1. ERD



3-2. 테이블 구조

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 명 | | Afps\_member | | SYSTEM | | AFPS groupaware | | |
| Entity 명 | | 사원 테이블 | | 작성자 | | 김은중 | | |
| Entity 정의 | | 사원, 부서, 직급에 관련된 정보를 관리하는 테이블 | | | | | | |
| NO | 한글명 | | Column | Key | 데이터 타입 | | 널 허용 | 비고 |
| 1 | 사원 인덱스 | | TIDX | PK | INTEGER | | NOT NULL | 사원 인덱스 |
| 2 | 사원 아이디 | | T\_ID |  | VARCHAR2(50) | | NOT NULL | 사원 아이디 |
| 3 | 사원 비밀번호 | | T\_PWD |  | VARCHAR2(200) | | NOT NULL | 사원 비밀번호 |
| 4 | 사원 이메일 | | T\_EMAIL |  | VARCHAR2(80) | | NOT NULL | 사원 이메일 |
| 5 | 사원 우편번호 | | T\_ADDR\_ZIPCODE |  | VARCHAR2(20) | | NOT NULL | 사원 우편번호 |
| 6 | 사원 일반주소 | | T\_ADDR\_GENERAL |  | VARCHAR2(80) | | NOT NULL | 사원 일반주소 |
| 7 | 사원 상세주소 | | T\_ADDR\_DETAIL |  | VARCHAR2(80) | | NOT NULL | 사원 상세주소 |
| 8 | 사원 연락처 | | T\_PHONE |  | VARCHAR2(50) | | NOT NULL | 사원 연락처 |
| 9 | 사원 이름 | | T\_NAME |  | VARCHAR2(20) | | NOT NULL | 사원 이름 |
| 10 | 사원 생년월일 | | T\_BIRTH |  | VARCHAR2(30) | | NOT NULL | 사원 생년월일 |
| 11 | 사원 직급 | | T\_POSITION |  | VARCHAR2(20) | | NOT NULL | 사원 직급 |
| 12 | 사원 부서 | | T\_DEPARTMENT |  | VARCHAR2(20) | | NOT NULL | 사원 부서 |
| 13 | 사원 입사일 | | T\_JOIN\_DATE |  | DATE | | NOT NULL | 사원 입사 날짜 |
| 14 | 사원 발생연차 | | T\_LEAVE\_GET |  | NUMBER | | NULL | 사원 기본 연차 |
| 15 | 사원 권한 | | T\_GRADE |  | VARCHAR2(1) | | NOT NULL | Z:일반권한, A:관리자권한 |
| 16 | 사원 퇴사 유무 | | DELYN |  | CHAR(1) | | NULL | Y:사원퇴사완료, N:사원퇴사미완료 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table 명 | | ANNUAL\_LEAVE | | SYSTEM | | AFSD groupaware | |
| Entity 명 | | 연차 관리 테이블 | | 작성자 | | 김은중 | |
| Entity 정의 | | 사용자 본인의 연차를 관리하는 테이블. | | | | | |
| NO | 한글명 | | Column | Key | 데이터 타입 | 널 허용 | 비고 |
| 1 | 연차 관리 인덱스 | | AIDX | PK  FK | INTEGER | NOT NULL | 연차 관리 인덱스 |
| 2 | 연차 유형 | | A\_TYPE |  | VARCHAR2(10) | NOT NULL | 연차 유형 |
| 3 | 비상 연락망 | | FRIEND\_PHONE |  | VARCHAR2(20) | NOT NULL | 비상 연락망 |
| 4 | 신청인 관계 | | FRIEND\_NAME |  | VARCHAR2(20) | NOT NULL | 신청인 관계 |
| 5 | 사용일 | | A\_USEDDAYS |  | NUMBER | NOT NULL | 사용일 |
| 8 | 사원 인덱스 | | TIDX | FK | INTEGER | NOT NULL | 사원 인덱스 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table 명 | | COMMUTE\_MANAGEMENT | | SYSTEM | | AFSD groupaware | |
| Entity 명 | | 근태 관리 테이블 | | 작성자 | | 이가람 | |
| Entity 정의 | | 사용자 본인의 근태를 관리하는 테이블. | | | | | |
| NO | 한글명 | | Column | Key | 데이터 타입 | 널 허용 | 비고 |
| 1 | 근태 관리 인덱스 | | CIDX | PK | NUMBER | NOT NULL | 근태 관리 인덱스 |
| 2 | 근태 유형 | | C\_TYPE |  | VARCHAR2(20) | NOT NULL | 정상, 결근, 주말, 연차 |
| 3 | 근무일자 | | C\_DATE |  | DATE | NOT NULL | 근무일자 |
| 4 | 출근 시간 | | C\_STARTTIME |  | VARCHAR2(20) | NOT NULL | 출근 시간 |
| 5 | 퇴근 시간 | | C\_ENDTIME |  | VARCHAR2(20) | NOT NULL | 퇴근 시간, 기본값0 |
| 6 | 근무 시간 | | C\_WORKTIME |  | VARCHAR2(20) | NOT NULL | 근무 시간,  기본값 0 |
| 7 | 사원 인덱스 | | TIDX | FK | INTEGER | NOT NULL | 사원 인덱스 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 명 | | | BOARD | | | SYSTEM | | AFSD groupaware |
| Entity 명 | | | 통합게시판 테이블 | | | 작성자 | | 윤수진 |
| Entity 정의 | | | 공지사항, 일반 게시판을 통합하여 관리하는 테이블 | | | | | |
| NO | 한글명 | Column | | Key | 데이터 타입 | | 널 허용 | 비고 |
| 1 | 게시판 인덱스 | bidx | | PK | INTEGER | | NOT NULL | 게시판 인덱스 |
| 2 | 게시판 유형 | b\_type | |  | CHAR(1) | | NOT NULL | N:공지사항 G:일반 게시글 |
| 3 | 게시글 제목 | b\_title | |  | VARCHAR2(80) | | NOT NULL | 게시글 제목 |
| 4 | 게시글 내용 | b\_content | |  | VARCHAR2(2048) | | NOT NULL | 게시글 내용 |
| 5 | 게시글 작성일 | b\_writedate | |  | DATE | | NOT NULL | 게시글 작성일 |
| 6 | 게시글 조회수 | b\_hit | |  | INTEGER | | NOT NULL | 게시글 조회수 |
| 7 | 삭제 유무 | delYN | |  | CHAR(1) | | NOT NULL | Y:삭제됨 N:삭제되지 않음 |
| 8 | 사원 인덱스 | tidx | | FK | INTEGER | | NOT NULL | 사원 인덱스 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 명 | | REPLY | | SYSTEM | | | AFSD groupaware | |
| Entity 명 | | 댓글 테이블 | | 작성자 | | | 윤수진 | |
| Entity 정의 | | 게시글의 댓글을 관리하는 테이블 | | | | | | |
| NO | 한글명 | | Column | Key | 데이터 타입 | 널 허용 | | 비고 |
| 1 | 댓글 인덱스 | | ridx | PK | INTEGER | NOT NULL | | 댓글 인덱스 |
| 2 | 댓글 내용 | | r\_content |  | VARCHAR2(2048) | NOT NULL | | 댓글 내용 |
| 3 | 댓글 작성일 | | r\_writedate |  | DATE | NOT NULL | | 댓글 작성일 |
| 4 | 삭제 유무 | | delYN |  | CHAR(1) | NOT NULL | | Y:삭제됨 N:삭제되지 않음 |
| 5 | 사원 인덱스 | | tidx | FK | INTEGER | NOT NULL | | 사원 인덱스 |
| 6 | 게시판 인덱스 | | bidx | FK | INTEGER | NOT NULL | | 게시판 인덱스 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table 명 | | DATA\_LIBRARY | | SYSTEM | | AFSD groupaware | |
| Entity 명 | | 자료실 테이블 | | 작성자 | | 윤수진 | |
| Entity 정의 | | 자료실의 정보를 관리하는 테이블 | | | | | |
| NO | 한글명 | | Column | Key | 데이터 타입 | 널 허용 | 비고 |
| 1 | 자료실 인덱스 | | didx | PK | NUMBER | NOT NULL | 자료실 인덱스 |
| 2 | 게시글 제목 | | d\_title |  | VARCHAR2(80) | NOT NULL | 게시글 제목 |
| 3 | 게시글 작성일 | | d\_writedate |  | DATE | NOT NULL | 게시글 작성일 |
| 4 | 자료 다운로드 수 | | d\_download |  | NUMBER | NULL | 자료 다운로드 수 |
| 5 | 삭제 유무 | | delYN |  | CHAR(1) | NOT NULL | Y:삭제됨  N:삭제되지 않음 |
| 6 | 사원 인덱스 | | tidx | FK | NUMBER | NOT NULL | 사원 인덱스 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 명 | | FILE\_INFO | | SYSTEM | | AFSD groupaware | | |
| Entity 명 | | 파일 정보 테이블 | | 작성자 | | 윤수진 | | |
| Entity 정의 | | 각 테이블에서 사용되는 파일의 정보를 관리하는 테이블 | | | | | | |
| NO | 한글명 | | Column | Key | 데이터 타입 | | 널 허용 | 비고 |
| 1 | 파일 인덱스 | | Fidx | PK | NUMBER | | NOT NULL | 파일 인덱스 |
| 2 | 파일 유형 | | f\_type |  | VARCHAR2(20) | | NOT NULL | 파일 유형 |
| 3 | 파일 이름 | | f\_org\_file\_name |  | VARCHAR2(80) | | NOT NULL | 파일 이름 |
| 4 | 파일 경로 | | f\_stored\_file\_name |  | VARCHAR2(80) | | NOT NULL | 파일 경로 |
| 5 | 파일 용량 | | f\_file\_size |  | VARCHAR2(80) | | NOT NULL | 파일 용량 |
| 6 | 게시판 인덱스 | | delyn | FK | NUMBER | | NULL | 게시판 인덱스 |
| 7 | 전자결재 인덱스 | | tidx | FK | NUMBER | | NULL | 사원 인덱스 |
| 8 | 사원 인덱스 | | bidx | FK | NUMBER | | NULL | 게시판 인덱스 |
| 9 | 연차 인덱스 | | didx | FK | NUMBER | | NULL | 자료실 인덱스 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 명 | | MANAGER\_EMAIL | | | SYSTEM | AFSD groupaware | | |
| Entity 명 | | 관리자 이메일 테이블 | | | 작성자 | 김은중 | | |
| Entity 정의 | | 관리자가 이메일 발송함, 이메일 발송을 관리하는 테이블 | | | | | | |
| NO | 한글명 | | Column | Key | 데이터 타입 | | 널 허용 | 비고 |
| 1 | 메일 인덱스 | | midx | PK | INTEGER | | NOT NULL | 이메일 발송 인덱스 |
| 2 | 메일 타입 | | m\_type |  | VARCHAR2(20) | | NOT NULL | 이메일 수신자 메일주소 |
| 3 | 수신인 | | m\_address |  | VARCHAR2(40) | | NOT NULL | 이메일 발송일 |
| 4 | 메일 제목 | | m\_title |  | VARCHAR2(40) | | NOT NULL | 이메일 제목 |
| 5 | 메일 내용 | | m\_content |  | VARCHAR2(4000) | | NOT NULL | 이메일 내용 |
| 6 | 발송일 | | m\_senddate |  | DATE | | NOT NULL | Y: 목록삭제완료, N:목록삭제미완료 |
| 7 | 삭제 유무 | | delyn |  | VARCHAR(1) | | NOT NULL | 삭제 유무 |
| 8 | 사원 인덱스 | | tidx | FK | INTEGER | | NULL | 사원 인덱스 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 명 | | electronic\_approval | | | SYSTEM | | | AFSD groupaware | |
| Entity 명 | | 전자결재 테이블 | | | 작성자 | | | 김륜광 | |
| Entity 정의 | | 사내 전자결재 정보를 관리하고 결재자와 결재신청자 간의 결재기능을 관리하는 테이블. | | | | | | | |
| NO | 한글명 | | Column | Key | | 데이터 타입 | 널 허용 | | 비고 |
| 1 | 전자결재  인덱스 | | edix | PK | | INTEGER | NOT NULL | | 전자결재  인덱스 |
| 2 | 결재상태 | | e\_status |  | | VARCHAR2(20) | NOT NULL | | 결재상태 |
| 3 | 결재유형 | | e\_type |  | | VARCHAR2(20) | NOT NULL | | 결재유형 |
| 4 | 기안일자 | | e\_draftdate |  | | VARCHAR2(20) | NULL | | 기안일자 |
| 5 | 시행일자 | | e\_startday |  | | VARCHAR2(20) | NULL | | 시행일자 |
| 6 | 종료일자 | | e\_endday |  | | VARCHAR2(20) | NULL | | 종료일자 |
| 7 | 합의부서 | | e\_con |  | | VARCHAR2(20) | NULL | | 합의부서 |
| 8 | 제목 | | e\_texttitle |  | | VARCHAR2(80) | NULL | | 제목 |
| 9 | 내용 | | e\_testcontent |  | | VARCHAR2(2048) | NULL | | 내용 |
| 10 | 사원인덱스 | | tidx | FK | | INTEGER | NOT NULL | | 사원인덱스 |
| 11 | 삭제유무 | | delYn |  | | VARCHAR2(1) | NOT NULL | | 삭제유무 |
| 12 | 반려유무 | | e\_approvaloyn |  | | VARCHAR2(1) | NULL | | 반려유무 |
| 13 | 반려자 | | e\_approvalnoperson |  | | VARCHAR2(10) | NULL | | 반려자 |
| 14 | 반려사유 | | e\_reason |  | | VARCHAR2(50) | NULL | | 반려사유 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 명 | | approval\_line | | | SYSTEM | | | AFSD groupaware | |
| Entity 명 | | 결재라인 테이블 | | | 작성자 | | | 김륜광 | |
| Entity 정의 | | 직급별 결재권한을 부여하는 테이블 | | | | | | | |
| NO | 한글명 | | Column | Key | | 데이터 타입 | 널 허용 | | 비고 |
| 1 | 문서인덱스 | | eidx | PK  FK | | INTEGER | NOT NULL | | 일정인덱스 |
| 2 | 팀장 | | teamleader |  | | VARCHAR2(20) | NOT NULL | | 일정유형 |
| 3 | 과장 | | sectionhead |  | | VARCHAR2(20) | NOT NULL | | 제목 |
| 4 | 부장 | | departmenthead |  | | VARCHAR2(20) | NULL | | 시작일 |
| 5 | 대표 | | leader |  | | VARCHAR2(20) | NOT NULL | | 종료일 |
| 6 | 결재순서 | | status |  | | VARCHAR2(4) | NOT NULL | | 내용 |
| 7 | 사원인덱스 | | tidx | FK | | INTEGER | NOT NULL | | 사원인덱스 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 명 | | schedule\_management | | | SYSTEM | | AFSD groupaware | | |
| Entity 명 | | 일정 테이블 | | | 작성자 | | 김륜광 | | |
| Entity 정의 | | 사내 일정에 관한 정보를 관리하는 테이블. | | | | | | | |
| NO | 한글명 | | Column | Key | | 데이터 타입 | | 널 허용 | 비고 |
| 1 | 일정인덱스 | | sidx | PK | | INTEGER | | NOT NULL | 일정인덱스 |
| 2 | 일정유형 | | s\_type |  | | VARCHAR2(4) | | NOT NULL | 일정유형 |
| 3 | 제목 | | s\_title |  | | VARCHAR2(20) | | NOT NULL | 제목 |
| 4 | 시작일 | | s\_startdate |  | | VARCHAR2(20) | | NULL | 시작일 |
| 5 | 종료일 | | s\_enddate |  | | VARCHAR2(100) | | NULL | 종료일 |
| 6 | 내용 | | s\_content |  | | VARCHAR2(50) | | NOT NULL | 내용 |
| 7 | 삭제유무 | | delYn |  | | VARCHAR2(50) | | NULL | 삭제유무 |
| 8 | 사원인덱스 | | tidx | FK | | INTEGER | | NOT NULL | 사원인덱스 |

3-3. 디렉토리 구조

|  |  |  |  |
| --- | --- | --- | --- |
| 상위 | 하위 | 파일명 | 설명 |
| AFPS | admin | add.jsp | 사원 등록 페이지 |
| check.jsp | 관리자 등록 여부  체크 페이지 |
| info.jsp | 관리자용 사원 정보 페이지 |
| join.jsp | 관리자 등록 페이지 |
| list.jsp | 관리자용 사원 목록 페이지 |
| approval | documentApproval.Again.jsp | 재기안 페이지 |
| documentContents.jsp | 전자결재 상세보기 페이지 |
| documentListMain.jsp | 전자결재 목록 페이지 |
| documentListMy.jsp | 내 전자결재 목록 페이지 |
| documentModify.jsp | 전자결재 수정 페이지 |
| documentWrite.jsp | 전자결 재기안 작성 페이지 |
| board | boardList.jsp | 게시판 목록 페이지 |
| boardModify.jsp | 게시판 수정 페이지 |
| boardView.jsp | 게시판 상세 페이지 |
| boardWrite.jsp | 게시판 등록 페이지 |
| myBoardView.jsp | 게시판 상세(팝업) |
| commute | commute.jsp | 근태 관리 페이지 |
| data | dataList.jsp | 자료실 목록 페이지 |
| email | email.jsp | 관리자 이메일 작성 페이지 |
| list.jsp | 관리자 이메일 목록 페이지 |
| memberList.jsp | 메일발송용 사원 목록 페이지 |
| read.jsp | 메일 상세 페이지 |
| include | main\_modal.jsp | 자료실용 include |
| new\_main.jsp | 공통 include |
| leave | modifay.jsp | 연차 수정 페이지 |
| view.jsp | 연차 상세 페이지 |
| write.jsp | 연차 작성 페이지 |
| main | realMain.jsp | 메인 페이지 |
| member | checklogin.jsp | 로그인 체크 페이지 |
| emailRead.jsp | 사원용 메일 상세 페이지 |
| findPwd.jsp | 비밀번호 찾기 페이지 |
| list.jsp | 사원용 사원 목록 페이지 |
| login.jsp | 로그인 페이지 |
| loginPost.jsp | 로그인 처리 페이지 |
| modifyPwd.jsp | 비밀번호 변경 페이지 |
| myHome.jsp | 내정보 페이지 |
| mypage.jsp | 마이 페이지 |
| schedule | scheduleBoard.jsp | 일정관리 게시판 목록 페이지 |
| scheduleCalendar.jsp | 일정관리 캘린더 목록 페이지 |
| scheduleContents.jsp | 일정관리 상세 페이지 |
| scheduleModify.jsp | 일정관리 수정 페이지 |
| schedulePopup.jsp | 일정관리 등록 페이지 |

3-4. 패키지 구조

|  |  |  |
| --- | --- | --- |
| 패키지명 | 클래스명 | 설명 |
| Controller | AdminController.java | 관리자 컨트롤러 |
| ApprovalController.java | 전자결재 컨트롤러 |
| BoardController.java | 게시판 컨트롤러 |
| CalendarController.java | 일정관리 컨트롤러 |
| CommuteController.java | 근태관리 컨트롤러 |
| DataLibarayController.java | 자료실 컨트롤러 |
| EmailController.java | 관리자 이메일 컨트롤러 |
| LeaveController.java | 연차 컨트롤러 |
| RealMainController.java | 메인 컨트롤러 |
| ReplyController.java | 댓글 컨트롤러 |
| Service | AdminService.java | 관리자 서비스 인터페이스 |
| AdminServiceImpl.java | 관리자 서비스 |
| ApprovalLineService.java | 전자결재 라인 서비스 인터페이스 |
| ApprovalLineServiceImpl.java | 전자결재 라인 서비스 |
| ApprovalService.java | 전자결재 서비스 인터페이스 |
| ApprovalServiceImpl.java | 전자결재 서비스 |
| BoardService.java | 게시판 서비스 인터페이스 |
| BoardServiceImpl.java | 게시판 서비스 |
| CommuteService.java | 근태관리 서비스 인터페이스 |
| CommuteServiveImpl.java | 근태관리 서비스 |
| DataLibraryService.java | 자료실 서비스 인터페이스 |
| DataLibraryServiceImpl.java | 자료실 서비스 |
| EmailService.java | 관리자 이메일 서비스 인터페이스 |
| EmailServiceImpl.java | 관리자 이메일 서비스 |
| FileInfoService.java | 파일 정보 서비스 인터페이스 |
| FileInfoServiceImpl.java | 파일 정보 서비스 |
| MyService.java | 내정보 서비스 인터페이스 |
| MyServiceImple.java | 내정보 서비스 |
| ReplyService.java | 댓글 서비스 인터페이스 |
| ReplyServiceImpl.java | 댓글 서비스 |
| ScheduleService.java | 일정관리 서비스 인터페이스 |
| ScheduleServiceImpl.java | 일정관리 서비스 |
| dao | AdminDAO.java | 관리자 DAO 인터페이스 |
| AdminDaoImpe.java | 관리자 DAO |
| ApprovaDAO.java | 전자결재 DAO 인터페이스 |
| ApprovaDAOlmpl.java | 전자결재 DAO |
| ApprovalLineDAO.java | 전자결재 라인 DAO 인터페이스 |
| ApprovalLineDAOImpl.java | 전자결재 라인DAO |
| BoardDAO.java | 게시판 DAO 인터페이스 |
| BoardDAOImpl.java | 게시판 DAO |
| CommuteDAO.java | 근태관리 DAO 인터페이스 |
| CommuteDAOImpl.java | 근태관리DAO |
| DataLibraryDAO.java | 자료실 DAO인터페이스 |
| DataLibraryDAOImpl.java | 자료실 DAO |
| EmailDAO.java | 관리자 이메일 DAO인터페이스 |
| EmailDAOImpl.java | 관리자 이메일 DAO |
| FileInfoDAO.java | 파일 정보 DAO 인터페이스 |
| FileInfoDAOImpl.java | 파일 정보DAO |
| MemberDAO.java | 사원 DAO인터페이스 |
| MemberDAOImpl.java | 사원DAO |
| MyDAO.java | 내정보 DAO 인터페이스 |
| MyDAOImpl.java | 내정보 DAO |
| ReplyDAO.java | 댓글 DAO 인터페이스 |
| ReplyDAOImpl.java | 댓글DAO |
| ScheduleDAO.java | 일정관리 DAO인터페이스 |
| ScheduleDAOImpl.java | 일정관리 DAO |

**4. 어플리케이션**

4-1. 주요 기능 및 정리

**i. 게시글 파일 다중 업로드**

form 태그를 통하여 게시글을 등록함. 글 제목, 내용을 입력할 수 있으며 파일 첨부 또한 가능함. 기본 1개의 파일을 첨부할 수 있으며 원하는 만큼 추가할 수 있음.

**view page > boardWrite.jsp**

<form id="writeAction" action="${pageContext.request.contextPath}/board/writeAction" method="post" enctype="multipart/form-data">

<%

Integer tidx = (Integer)session.getAttribute("userTidx");

String grade = (String)session.getAttribute("userGrade");

%>

<input type="hidden" name="grade" value="<%=grade%>">

<input type="hidden" name="tidx" value="<%=tidx%>">

<table id="write" class="table">

<tr>

<th style="width:25%;">글제목</th>

<td>

<input type="text" class="form-control" name="b\_title" id="title">

</td>

</tr>

<tr>

<th style="width:25%;">내용</th>

<td>

<textarea name="b\_content" id="content" class="form-control"></textarea>

</td>

</tr>

<tr>

<th style="width:25%;">첨부파일</th>

<td style="text-align:left;">

<div id="fileArea">

<input type='file' name='file\_0'>

</div>

</td>

</tr>

</table>

<div id="btnArea">

<button type="button" class="btn btn-primary btn-sm float-right" onclick="location.href='${pageContext.request.contextPath}/board/list'">

취소<button>

<button type="button" class="btn btn-primary btn-sm" id="fileAdd">파일추가</button>

<button type="button" class="btn btn-primary btn-sm float-right" id="submitBtn">등록</button>

</div>

</form>

파일 추가 버튼을 클릭하면 append 함수를 통해 파일 선택 버튼이 추가적으로 생성됨. 이때 fileCount를 두어 파일 버튼마다 각각 id 값을 다르게 지정해줌.

**view page > boardWrite.jsp > javascript**

$(document).ready(function(){

fileCount = 1;

$("#fileAdd").on("click", function(){

$("#fileArea").append("<div><input type='file' name='file\_"+(fileCount++)+"'/>"

+"<button type='button' style='float:right;' id='fileDelBtn'>"+"삭제"+"</button></div>");

$(document).on("click","#fileDelBtn", function(){

$(this).parent().remove();

});

});

});

**Controller > boardController**

@RequestMapping(value = "/writeAction")

public String writeAction(BoardVO vo, String grade, HttpServletRequest request,

MultipartHttpServletRequest mpRequest) throws Exception {

logger.info("BoardWriteAction");

BoardVO bvo = vo;

if (grade.equals("Z")) {

bvo.setB\_type("G");

service.writeBoard(bvo, mpRequest);

return "redirect:/board/list";

} else if (grade.equals("A")) {

bvo.setB\_type("N");

service.writeBoard(bvo, mpRequest);

return "redirect:/board/list";

}

String referer = request.getHeader("Referer");

return "redirect:" + referer;

}

**Service > boardService**

public void writeBoard(BoardVO vo, MultipartHttpServletRequest mpRequest) throws Exception;

boardServiceImpl에서는 fileUtils의 parseIntertFileInfoBoard 메소드를 호출. List<Map<String, Object>> 형식으로 이를 받아옴. 이후 boardDAO의 insertFile 메소드에 List 안의 Map을 하나씩 넘겨줌.

**Service > boardServiceImpl**

@Override

public void writeBoard(BoardVO vo, MultipartHttpServletRequest mpRequest) throws Exception {

vo.setB\_content(removeHtml.removeHtml(vo.getB\_content()));

dao.writeBoard(vo);

List<Map<String, Object>> list = fileUtils.parseInsertFileInfoBoard(vo, mpRequest);

int size = list.size();

for (int i = 0; i < size; i++) {

dao.insertFile(list.get(i));

}

}

RequestContextHolder를 통해 Request에 대한 정보를 가져옴. 이후 webapp 폴더의 실제 경로를 구함. getFileNames를 통해 파일의 이름을 구하고 Iterater 형태로 넣어줌. Iterater로 while문을 돌며 파일들의 정보를 가져오고 이를 List<Map<String, Object>>형식으로 저장해 리턴함.

**Utils> FileUtils**

public List<Map<String, Object>> parseInsertFileInfoBoard(BoardVO boardVO,

MultipartHttpServletRequest mpRequest) throws Exception{

HttpServletRequest request = ((ServletRequestAttributes)RequestContextHolder

.getRequestAttributes()).getRequest();

String filePath = request.getSession().getServletContext().getRealPath("/resources/static/file/");

Iterator<String> iterator = mpRequest.getFileNames();

MultipartFile multipartFile = null;

String originalFileName = null;

String originalFileExtension = null;

String storedFileName = null;

List<Map<String, Object>> list = new ArrayList<Map<String,Object>>();

Map<String, Object> listMap = null;

int bidx = boardVO.getBidx();

File file = new File(filePath);

if(file.exists() == false) {

file.mkdirs();

}

while(iterator.hasNext()) {

multipartFile = mpRequest.getFile(iterator.next());

if(multipartFile.isEmpty() == false) {

originalFileName = multipartFile.getOriginalFilename();

originalFileExtension = originalFileName.substring(originalFileName.lastIndexOf("."));

storedFileName = getRandomString() + originalFileExtension;

file = new File(filePath + storedFileName);

multipartFile.transferTo(file);

String extension = StringUtils.getFilenameExtension(originalFileName);

listMap = new HashMap<String, Object>();

listMap.put("bidx", bidx);

listMap.put("F\_TYPE", extension);

listMap.put("ORG\_FILE\_NAME", originalFileName);

listMap.put("STORED\_FILE\_NAME", storedFileName);

listMap.put("FILE\_SIZE", multipartFile.getSize());

list.add(listMap);

}

}

return list;

}

**DAO > boardDAO**

public void writeBoard(BoardVO vo) throws Exception;

public void insertFile(Map<String, Object> map) throws Exception;

**DAO > boardDAOImpl**

@Override

public void writeBoard(BoardVO vo) throws Exception {

sqlSession.insert(Namespace+".writeBoard", vo);

}

@Override

public void insertFile(Map<String, Object> map) throws Exception {

sqlSession.insert(Namespace+".insertFile", map);

}

**Mapper > boardMapper**

<insert id="writeBoard" parameterType="boardVO"

useGeneratedKeys="true" keyProperty="bidx">

INSERT INTO BOARD(

B\_TYPE,

B\_TITLE,

B\_CONTENT,

B\_WRITEDATE,

B\_HIT,

DELYN,

TIDX

) VALUES(

#{b\_type},

#{b\_title},

#{b\_content},

now(),

0,

'N',

#{tidx}

)

</insert>

<insert id="insertFile" parameterType="hashMap">

INSERT INTO FILE\_INFO(

F\_TYPE,

F\_ORG\_FILE\_NAME,

F\_STORED\_FILE\_NAME,

F\_FILE\_SIZE,

DELYN,

BIDX

)VALUES(

#{F\_TYPE},

#{ORG\_FILE\_NAME}

#{STORED\_FILE\_NAME},

#{FILE\_SIZE},

'N',

#{bidx}

)

</insert>

**ii. 부서별 사원 리스트 출력**

**view page > List.jsp**

부서별 리스트를 제공하고 사용자가 원하는 부서 선택

<select id="t\_department" name="t\_department" class="selectBuseo form-select" style="height:30px; font-size:0.5rem;">

<option value="all">::부서::</option>

<option value="마케팅">마케팅</option>

<option value="판매">판매</option>

<option value="디자인">디자인</option>

<option value="인사">인사</option>

<option value="재정">재정</option>

<option value="회계">회계</option>

</select>

**view page > List.jsp > javascript**

부서 선택하는 태그에 아이디값을 줘서 option이 변경될 때마다 이벤트 실행. 변경된 option의 벨류값을 부서별 사원리스트를 조회하는 가상주소로 data 전송ajax통신 후 받아온 데이터를 변수에 저장. 사원리스트를 보여주는 태그에 해당 변수 추가

$("#t\_department").on("change", function(){

$.ajax({

url:$("#domain").val()+"/member/buseolist",

data: $("#t\_department").serialize(),

dataType:"json",

success:function(searchDepartmentMember){

var solist = "";

if(searchDepartmentMember.length>0){

for(var i=0; i<searchDepartmentMember.length;i++){

solist += "<div class='info'>";

solist += "<input type='hidden' id='tidx' name='tidx' value='"+searchDepartmentMember[i].tidx+"'>";

if(searchDepartmentMember[i].f\_stored\_file\_name!=null){

solist += "<img style='width:60px;height:40px;border:1px solid lightgray;' src='<c:url value='/resources/static/profile/"+searchDepartmentMember[i].f\_stored\_file\_name+"'/>' alt='프로필이미지'></td>";

}else{

solist += "<img src='<c:url value='/resources/static/profile/notprofile.jpg'/>' class='listProfile' alt='프로필이미지'>";

}

solist += " <span>";

solist += searchDepartmentMember[i].t\_name;

solist += "</span>";

solist += " <span>";

solist += searchDepartmentMember[i].t\_department;

solist += "</span>";

solist += " <span>";

solist += searchDepartmentMember[i].t\_position;

solist += "</span>";

solist += " <span>";

solist += searchDepartmentMember[i].t\_email;

solist += "</span>";

solist += "</div>";

}

$("#tbodyx").html("<tr></tr>");

$(".saone").html(solist);

}else{

solist += "<span>";

solist += "해당부서는 사원이 없습니다.";

solist += "</span>";

$("#tbodyx").html("<tr></tr>");

$(".saone").html(solist);

}

}

});

});

**Controller > memberController**

Ajax 요청으로 받아온 데이터를 이용해 부서가 선택이 되어있는지 유무 체크 후 서비스 메서드에 매개변수로 부서이름을 넘기고 호출. 받아온 값을 리스트에 저장. 저장된 리스트를 요청이 들어온 곳에 반환.

@RequestMapping(value = "/buseolist")

@ResponseBody

public Object searchDepartmentMember(Locale locale, Model model, String t\_department,HttpServletRequest request) throws Exception {

logger.info("부서별 사원리스트", locale);

if(t\_department!=null) {

if(t\_department.equals("all")) {

List<MemberDTO> searchDepartmentMember = service.selectAllMember2();

model.addAttribute("searchDepartmentMember",searchDepartmentMember);

return searchDepartmentMember;

}

List<MemberDTO> searchDepartmentMember = service.searchDepartmentMember(t\_department);

model.addAttribute("searchDepartmentMember",searchDepartmentMember);

return searchDepartmentMember;

}

Return "<script>location.href='"+request.getContextPath()+"/member/login';</script>";

}

**Service > memberService**

memberServiceImpl에 부서별 리스트를 보는 메서드 호출

//부서별 검색

public List<MemberDTO> searchDepartmentMember(@Param(value="t\_department")String t\_department) throws Exception;

**Service > memberServiceImpl**

memberDAO 에 부서별 리스트를 보는 메서드 호출

//부서별 검색

@Override

public List<MemberDTO> searchDepartmentMember(String t\_department)throws Exception{

return dao.searchDepartmentMember(t\_department);

}

**DAO > memberDAO**

memberDAOImpl 에 부서별 리스트를 보는 메서드 호출

public List<MemberDTO> searchDepartmentMember(String t\_department) throws Exception;

**DAO > memberDAO Impl**

sqlSession에서 리스트를 받는 메서드 호출 매개변수로 해당 sql문을 가지고 있는 mapper의 경로와 id값 및 받아온 부서의 이름을 같이 넘김

@Override

public List<MemberDTO> searchDepartmentMember(String t\_department) throws Exception{

return sqlSession.selectList(Namespace+".searchDepartmentMember",t\_department);

}

**Mapper > memberMapper**

<select id="searchDepartmentMember" resultType="memberDTO">

SELECT A.\*,B.F\_STORED\_FILE\_NAME,B.F\_TYPE

FROM AFPS\_MEMBER A LEFT JOIN FILE\_INFO B ON A.TIDX = B.TIDX WHERE A.T\_DEPARTMENT = #{t\_department} AND A.DELYN = 'N'

</select>

**iii. 퇴근 등록**

메인페이지에 위치한 근태관리 위젯의 퇴근 버튼 클릭.

**View Page > realMain.jsp**

<div id="box3" class="inner\_content" style="width: 100%; height: 100%; font-size: 50px;">

<div class="conTitle">근태관리</div>

<div class="conCon"style="display: grid; grid-template-columns: 1fr 1fr;">

  <input type="hidden" id="tidx" value="<%=userTidx%>">

<button style="grid-columns: 1/2; margin: 0 5px;" id="startWork" type="button" class="btn btn-outline-primary btn-lg float-right" onclick="location.href='${pageContext.request.contextPath}/commute/startCommute?tidx=<%=userTidx%>'">출근</button>

<button style="grid-columns: 2/3; margin: 0 5px;" id="endWork" type="button"

class="btn btn-outline-danger btn-lg float-right" onclick="location.href='${pageContext.request.contextPath}/commute/endCommute?tidx=<%=userTidx%>'">퇴근</button>

</div>

</div>

컨트롤러에서 오늘 날짜의 퇴근 기록이 있다면 ‘이미 퇴근처리되었습니다’라는 알림창과 함께 메인페이지로 돌아가고, 기록이 없다면 퇴근처리가 됨.

**Controller > CommuteController.java**

@Controller

public class CommuteController {

private static final Logger logger = LoggerFactory.getLogger(CommuteController.class);

@Inject

private CommuteService service;

@RequestMapping(value = "/commute/endCommute")

public String endCommute(@RequestParam Map<String, String> param,

HttpServletResponse   response, HttpServletRequest request) throws Exception {

logger.info("endCommute...");

int tidx = Integer.parseInt(param.get("tidx"));

response.setContentType("text/html; charset=utf-8");

PrintWriter out = response.getWriter();

boolean test = service.addEndWork(tidx, response);

if(test) {

out.println("<script>alert('이미 퇴근 처리되었습니다.');</script>");

out.println("<script>window.location.href='"+request.getContextPath()

+"/main/mainPage';</script>");

out.close();

}else {

out.println("<script>alert('퇴근 처리되었습니다.');</script>");

out.println("<script>window.location.href='"+request.getContextPath()

+"/main/mainPage';</script>");

out.close();

}

return "redirect:/main/mainPage";

}

}

**Service > CommuteService.java**

public interface CommuteService {

public boolean addEndWork(int tidx, HttpServletResponse response) throws Exception;

public String selectStartWork(int tidx) throws Exception;

public String selectEndWork(int tidx) throws Exception;

}

현재 날짜를 포맷해서 퇴근 시간으로 등록하고, 오늘 날짜의 출근 시간을 조회해서 분으로 변환한 후 퇴근시간과 출근시간의 차를 구해 근무시간으로 저장함. 현재 날짜의 출근 기록이 없는데 퇴근 버튼을 클릭하면 ‘출근 기록이 없습니다.’라는 알림창이 나오고 메인으로 돌아감.

**Service > CommuteServiceImpl.java**

@Service

public class CommuteServiceImpl implements CommuteService{

@Inject

private CommuteDAO dao;

//퇴근 등록

@Override

public boolean addEndWork(int tidx, HttpServletResponse response) throws Exception {

try {

Date today = new Date();

SimpleDateFormat format2 = new SimpleDateFormat("HH:mm");

String endTime = format2.format(today);

CommuteVO vo = new CommuteVO();

vo.setC\_endtime(endTime);

vo.setTidx(tidx);

if(dao.selectEndWork(tidx).equals("0")) {

String start = dao.selectStartWork(tidx);

int stime1 = Integer.parseInt(start.split(":")[0]);

int stime2 = Integer.parseInt(start.split(":")[1]);

int stimeresult = (stime1\*60)+stime2;

Calendar cal = Calendar.getInstance();

int hour = cal.get(Calendar.HOUR\_OF\_DAY);

int min = cal.get(Calendar.MINUTE);

int etimeresult = (hour\*60)+min;

String result = (etimeresult - stimeresult)+"";

vo.setC\_worktime(result);

dao.addEndWork(vo);

return false;

}

else {

return true;

}

}catch(Exception e) {

response.setContentType("text/html; charset=utf-8");

PrintWriter out = response.getWriter();

out.println("<script>alert('출근기록이　　　　　　　　　　　　　없습니다.');history.back();</script>");

out.close();

}

return false;

}

//출근 기록 조회

@Override

public String selectStartWork(int tidx) throws Exception {

return dao.selectStartWork(tidx);

}

//퇴근 기록 조회

@Override

public String selectEndWork(int tidx) throws Exception {

return dao.selectEndWork(tidx);

}

}

**DAO > CommuteDAO.java**

public interface CommuteDAO {

public void addEndWork(CommuteVO vo) throws Exception;

public String selectStartWork(int tidx) throws Exception;

public String selectEndWork(int tidx) throws Exception;

}

**DAO > CommuteDAOImpl.java**

@Repository

public class CommuteDAOImpl implements CommuteDAO{

@Inject

private SqlSession sqlSession;

private static final String Namespace = "tobe.project.mapper.commuteMapper";

//퇴근 등록

@Override

public void addEndWork(CommuteVO vo) throws Exception {

sqlSession.update(Namespace+".addEndWork", vo);

}

//출근 기록 조회

@Override

public String selectStartWork(int tidx) throws Exception {

return sqlSession.selectOne(Namespace+".selectStartWork", tidx);

}

//퇴근 기록 조회

@Override

public String selectEndWork(int tidx) throws Exception {

return sqlSession.selectOne(Namespace+".selectEndWork", tidx);

}

}

**mappers > commuteMapper.xml**

<mapper namespace="tobe.project.mapper.commuteMapper">

<!-- 퇴근시간 등록 -->

<update id="addEndWork" parameterType="CommuteVO">

update commute\_management set c\_endtime = #{c\_endtime}, c\_worktime = #{c\_worktime}

where tidx = #{tidx} and c\_date = curdate()

</update>

<!-- 출근기록 조회 -->

<select id="selectStartWork" resultType="String">

select c\_starttime from (

select \* from commute\_management

order by cidx desc

)tb\_commute where tidx = #{tidx} and c\_date = curdate()

</select>

<!-- 퇴근기록 조회 -->

<select id="selectEndWork" resultType="String">

select c\_endtime from (

select \* from commute\_management

order by cidx desc

)tb\_commute where tidx = #{tidx} and c\_date = curdate()

</select>

</mapper>

근태관리페이지에서 근무기록 리스트로 출력.

**View Page > commute.jsp**

<script>

function selectAllCommute(list){

var c\_list = "";

if(list.length > 0){

for(var i=0; i<list.length; i++){

c\_list += "<tr>";

c\_list += "<td>";

c\_list += list[i].c\_type;

c\_list += "</td>";

c\_list += "<td>";

if(list[i].c\_date != null){

var cdate = list[i].c\_date;

var result = cdate.substring(0,10);

c\_list += result;

}

c\_list += "<input type='hidden' value='"+list[i].cidx+"'id='cidx' name='cidx'>";

c\_list += "</td>";

c\_list += "<td id='starttime'>";

c\_list += list[i].c\_starttime;

c\_list += "</td>";

c\_list += "<td>";

c\_list += list[i].c\_endtime;

c\_list += "</td>";

c\_list += "<td>";

c\_list += list[i].c\_worktime;

c\_list += "</td>";

c\_list += "</tr>";

}

$("#c\_list").html(c\_list);

}

}

</script>

<table class="table">

<thead>

<tr>

<th>근태유형</th>

<th>출퇴근일자</th>

<th>출근기록</th>

<th>퇴근기록</th>

<th>근무시간(분)</th>

</tr>

</thead>

<tbody id="c\_list">

<c:forEach items="${list}" var="list">

<tr>

<td>${list.c\_type}</td>

<td>

<c:set var="cdate" value="${list.c\_date}"/>

${fn:substring(cdate,0,10)}

<input type="hidden" value="${list.cidx}" id="cidx" name="cidx">

</td>

<td id="starttime">${list.c\_starttime}</td>

<td>${list.c\_endtime}</td>

<td>${list.c\_worktime}</td>

</tr>

</c:forEach>

</tbody>

</table>

**iiii. 결재 최종승인시 캘린더 자동등록**

대표가 최종승인하면 결재문서번호를 ajax를 이용해 컨트롤러로 전송

컨트롤러에서　결재문서번호를　이용해　데이터　검색

검색한　문서　결재　완료로　업데이트하고　결재문서의　타입이　휴가결재문서가　아니면 　기안유형에　따라　일정등록　 기안문서가　아니고　휴가종류이면　데이터의　ｔｉｄｘ로　사원정보조회　사원의　잔여연차에서　사용한연차빼고　사원정보　업데이트　개인일정으로　일정등록

**DocumentContents.jsp**

<%if(userTidx != mo.getTidx() && userPosition.equals("대표") && userId.equals(to.getLeader()) && to.getStatus().equals("0003")){ %>

<button type="button" class="btn btn-primary btn-sm float-right" onclick="ok2()">최종승인</button>

<button type="button" class="btn btn-outline-danger btn-sm" id="no">반려</button>

<button type="button" class="btn btn-primary btn-sm float-right" onclick="window.close();">닫기</button>

<%}%>

**DocumentContents.jsp > ajax**

**function** ok2(){

Swal.fire({

title:"최종 승인",

text:"최종 승인 하시겠습니까?",

icon:"question",

showCancelButton:**true**,

confirmButtonText:"확인",

cancelButtonText:"취소"

}).then(result => {

**if**(result.isConfirmed){

$.ajax({

url: "<%=request.getContextPath()%>/approval/documentOk?eidx=<%=contents.getEidx()%>",

dataType:'json',

success:**function**(){

Swal.fire("결재 승인","최종승인 처리되었습니다.","success").then(result => {

opener.parent.location.reload();

window.close();

});

},

error:**function**(){

Swal.fire("결재 승인","최종승인 실패되었습니다.","error").then(result => {

opener.parent.location.reload();

window.close();

});

}

})

}

});

}

**ApprovalController**

@ResponseBody

@RequestMapping(value = "/documentOk")

**public** ApprovalDTO documentOK(Model model,**int** eidx) **throws** Exception{

ApprovalDTO to = service.selectOneApprovalDocumentContents(eidx);

ApprovalDTO to2 = lservice.selectOneApprovalLine(eidx);

LeaveDTO dto = myservice.selectOneLeave(eidx);

**int** approvalOk = service.modifyApprovalStatusOk(eidx);

**if**{

lservice.modifyApprovalLeader(eidx);

**if**(!to.getE\_type().equals("연차")&& !to.getE\_type().equals("월차")&& !to.getE\_type().equals("병가")&& !to.getE\_type().equals("반차")){

**if**(approvalOk == 1){

ScheduleVO vo = **new** ScheduleVO();

vo.setS\_type(to.getE\_type());

vo.setS\_title(to.getE\_textTitle());

vo.setS\_startDate(to.getE\_startDay());

vo.setS\_endDate(to.getE\_endDay());

vo.setS\_content(to.getE\_textContent());

vo.setTidx(to.getTidx());

sservice.addSchedule(vo);

}

}**else** {

MemberDTO memb = mservice.selectOneMemberIdx(to2.getTidx());

**double** checkLeave = memb.getT\_leave\_get() - dto.getA\_useddays();

dto.setA\_useddays(checkLeave);

**int** mySchedule = myservice.updateLeave(dto);

**if**(mySchedule == 1) {

ScheduleVO vo = **new** ScheduleVO();

vo.setS\_type(dto.getE\_type());

vo.setS\_title(dto.getE\_texttitle());

vo.setS\_startDate(dto.getE\_startday());

vo.setS\_endDate(dto.getE\_endday());

vo.setS\_content(dto.getE\_textcontent());

vo.setTidx(dto.getTidx());

sservice.addSchedule(vo);

}

}

}

**return** to2;

}

**ApprovalDAO**

**public** ApprovalDTO selectOneApprovalDocumentContents (**int** eidx) **throws** Exception;

**public** **int** modifyApprovalStatusOk(**int** eidx) **throws** Exception;

**ApprovalDAOImpl**

@Override

**public** ApprovalDTO selectOneApprovalDocumentContents(**int** eidx) **throws** Exception {

**return** sqlSession.selectOne(***Namespace***+".selectOneApprovalDocumentContents",eidx);

}

@Override

**public** **int** modifyApprovalStatusOk(**int** eidx) **throws** Exception {

**return** sqlSession.update(***Namespace***+".modifyApprovalStatusOk",eidx);

}

**ApprovalService**

**public** ApprovalDTO selectOneApprovalDocumentContents(**int** eidx) **throws** Exception;

**public** **int** modifyApprovalStatusOk(**int** eidx) **throws** Exception;

**ApprovalServiceImpl**

@Override

**public** ApprovalDTO selectOneApprovalDocumentContents(**int** eidx) **throws** Exception {

**return** dao.selectOneApprovalDocumentContents(eidx);

}

@Override

**public** **int** modifyApprovalStatusOk(**int** eidx) **throws** Exception {

**return** dao.modifyApprovalStatusOk(eidx);

}

**ApprovalLineDAO**

**public** ApprovalDTO selectOneApprovalLine(**int** eidx) **throws** Exception;

**public** **void** modifyApprovalLeader(**int** eidx) **throws** Exception;

**ApprovalLineDAOImpl**

@Override

**public** ApprovalDTO selectOneApprovalLine(**int** eidx) **throws** Exception {

**return** sqlSession.selectOne(***Namespace***+".selectOneApprovalLine",eidx);

}

@Override

**public** **void** modifyApprovalLeader(**int** eidx) **throws** Exception {

sqlSession.update(***Namespace***+".modifyApprovalLeader",eidx);

}

**ApprovalLineService**

**public** ApprovalDTO selectOneApprovalLine(**int** eidx) **throws** Exception;

**public** **void** modifyApprovalLeader(**int** eidx) **throws** Exception;

**ApprovalLineServiceImpl**

@Override

**public** ApprovalDTO selectOneApprovalLine(**int** eidx) **throws** Exception {

**return** dao.selectOneApprovalLine(eidx);

}

@Override

**public** **void** modifyApprovalLeader(**int** eidx) **throws** Exception {

dao.modifyApprovalLeader(eidx);

}

MyDAO

**public** LeaveDTO selectOneLeave(**int** eidx) **throws** Exception;

**MyDAOImpl**

@Override

**public** LeaveDTO selectOneLeave(**int** eidx) **throws** Exception {

**return** sqlSession.selectOne(***Namespace***+".selectOneLeave",eidx);

}

MyService

**public** LeaveDTO selectOneLeave(**int** eidx) **throws** Exception;

MyServiceImpl

@Override

**public** LeaveDTO selectOneLeave(**int** eidx) **throws** Exception {

**return** dao.selectOneLeave(eidx);

}

**ScheduleDAO**

**public** **void** addSchedule(ScheduleVO vo) **throws** Exception;

**ScheduleDAOImpl**

@Override

**public** **void** addSchedule(ScheduleVO vo) **throws** Exception{

sqlSession.insert(***Namespace***+".addSchedule",vo);

}

**ScheduleService**

**public** **void** addSchedule(ScheduleVO vo) **throws** Exception;

**ScheduleServiceImpl**

@Override

**public** **void** addSchedule(ScheduleVO vo) **throws** Exception{

dao.addSchedule(vo);

}

**MemberDAO**

**public** MemberDTO selectOneMemberIdx(**int** tidx) **throws** Exception;

**MemberDAOImpl**

@Override

**public** MemberDTO selectOneMemberIdx(**int** tidx) **throws** Exception {

**return** sqlSession.selectOne(***Namespace***+".selectOneMemberIdx", tidx);

}

**MemberService**

**public** MemberDTO selectOneMemberIdx(@Param(value="tidx")**int** tidx) **throws** Exception;

**MemberServiceImpl**

@Override

**public** MemberDTO selectOneMemberIdx(**int** tidx) **throws** Exception {

**return** dao.selectOneMemberIdx(tidx);

}

**ApprovalMapper**

<select id=*"selectOneApprovalDocumentContents"* resultType=*"ApprovalDTO"* parameterType=*"ApprovalDTO"*>

SELECT @rownum:=@rownum+1 rnum, EIDX, E\_TYPE, E\_DRAFTDATE, E\_STARTDAY, E\_ENDDAY, E\_CON, E\_TEXTTITLE, E\_TEXTCONTENT, E\_STATUS, TIDX, E\_APPROVALNOYN, E\_APPROVALNODAY, E\_APPROVALNOPERSON, E\_REASON, T\_NAME, T\_ID, T\_DEPARTMENT, T\_POSITION, TEAMLEADER, DEPARTMENTHEAD, SECTIONHEAD, LEADER, STATUS, DELYN

from(

SELECT

(SELECT @ROWNUM := 0) R, A.\*, B.T\_NAME, T\_ID, T\_DEPARTMENT, T\_POSITION,C.TEAMLEADER, DEPARTMENTHEAD, SECTIONHEAD, LEADER, STATUS

FROM

ELECTRONIC\_APPROVAL A left join AFPS\_MEMBER B ON A.TIDX = B.TIDX

LEFT JOIN APPROVAL\_LINE C on A.EIDX = C.EIDX

WHERE

A.DELYN = 'N'

)a

WHERE

eidx = #{eidx}

</select>

<update id=*"modifyApprovalStatusOk"*>

update

ELECTRONIC\_APPROVAL

set

E\_STATUS = '결재완료'

where

eidx = #{eidx

</update>

**ApprovalLineMapper**

<select id=*"selectOneApprovalLine"* resultType=*"ApprovalDTO"* parameterType=*"ApprovalDTO"*>

select \* from APPROVAL\_LINE where EIDX = #{eidx}

</select>

<update id=*"modifyApprovalLeader"* parameterType=*"ApprovalDTO"*>

UPDATE

APPROVAL\_LINE

SET

STATUS = '0000'

WHERE

EIDX = #{eidx}

</update>

**MyMapper**

<select id=*"selectOneLeave"* resultType=*"leaveDTO"*>

SELECT A.\*,B.\*,C.\* FROM ELECTRONIC\_APPROVAL A

LEFT JOIN APPROVAL\_LINE B ON A.EIDX = B.EIDX

INNER JOIN ANNUAL\_LEAVE C ON A.EIDX = C.EIDX

WHERE DELYN = 'N'

AND A.EIDX = #{eidx}

</select>

**ScheduleMapper**

<insert id=*"addSchedule"* parameterType=*"scheduleVO"*>

INSERT INTO

SCHEDULE\_MANAGEMENT(S\_TYPE, S\_TITLE, S\_STARTDATE, S\_ENDDATE, S\_CONTENT , tidx)

VALUES

(#{s\_type},#{s\_title},#{s\_startDate},#{s\_endDate},#{s\_content} ,#{tidx})

</insert>

**MemberMapper**

<select id=*"selectOneMemberIdx"* resultType=*"memberDTO"*>

SELECT A.\*,B.F\_STORED\_FILE\_NAME,B.F\_TYPE

FROM AFPS\_MEMBER A LEFT JOIN FILE\_INFO B ON A.TIDX = B.TIDX WHERE B.F\_TYPE = 'profile' AND A.TIDX = #{tidx}

<!--AND B.F\_TYPE = 'profile' -->

</select>

**5. 프로젝트 후기**

|  |  |
| --- | --- |
| 이름 | 후기 |
| 윤수진 | 처음 맡는 팀장이라는 자리에 부담도 걱정도 많았습니다. 그러나 모든 팀원께서 오히려 저를 이끌어주시며 힘을 북돋아주셨습니다. 당면한 문제게 다 함께 머리를 맞대고 함께 고민하고 의견을 나누며 서로가 서로의 디딤돌이 되어주었습니다. 기술적인 능력 이전에 팀워크가 있었기에 만족스러운 결과물이 나올 수 있었습니다. 이번 경험을 발판 삼아 개발자로서 더욱 발전해나가고자 합니다. |
| 김은중 | 이 프로젝트를 시작하면서 잘 할 수 있을지 걱정도 되고 처음 만나는 사람들과 잘 맞춰서 진행이 될지 생각이 많았다. 처음에는 말을 한번도 해본 적이 없어서 팀원들의 성향을 잘 몰라 의견이 많이 달라서 힘들었는데 시간이 지나면서 많은 이야기도 하고 서로 각자의 의견을 존중해주고 이해하게 되어 좋았다. 어떻게 보면 길고 짧은 시간 이였는데 처음이라서 그런지 취업을 해서도 자주 생각날 것 같다. |
| 이가람 | 프로젝트 시작 시 많은 걱정을 했었는데 팀장님이 잘 이끌어 주셔서 무사히 끝낼 수 있었습니다. 모르는 부분은 서로서로 도와주었는데, 프로젝트에서는 팀워크가 중요하다는 것을 다시 한 번 깨닫게 되었습니다. 또한 문제해결에 대한 방법과 보충해야 할 점도 알게 되었습니다. 마지막으로 힘들고 어려웠을 때도 있었지만 포기하지 않고 마무리를 지은 이번 경험으로 인해 다음 일의 시작을 너무 두려워하지 않을 것 같습니다.  좋은 경험이었고 도와주신 모든 분께 감사드립니다. |
| 김륜광 | 실무에 참여하게 되었을 때 팀원들의 의사소통 방식과 프로젝트를 진행해야 하는 부분에 있어서 직접적으로 몸소 경험할 수 있는 아주 좋은 기회였고 기술적으로 해볼 수 있는 많은 도전들을 시도해볼 수 있고 접목해볼 수 있어 좋았다.  아쉬운 점은 깃을 제대로 공부하지 못한 것이고, 코드의 재사용과 가독성에 대한 부분을 중요하게 생각하는 계기가 되었고, 프로젝트를 진행하면서 가장 잘 안되었던 부분이기도 했다.  하지만 많이 틀려봐야 그로 인해 얻는 것 또한 많아 지는 좋은 시간이었다. 욕심냈던 만큼 스스로에 대한 실망도 있지만 한층 더 개발자로서 토대를 다지는 계기가 되었다. |