

JAVA PROJECT

Workin App

근무 관리 시스템

TEAM

김도균
김태현
조현민

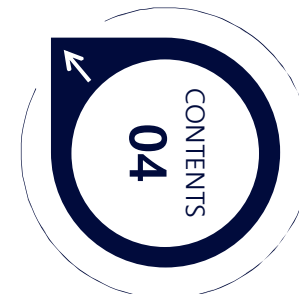
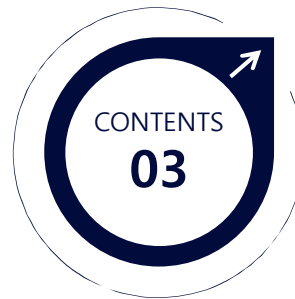
목차

프로젝트 목표



역할 분담

플로우차트



구현 기능 설명

시연



프로젝트 목표

- 1) 자바훈련과정을 진행하면서 습득한 기술을 모두 활용할 수 있는 응용프로그램 구현
- 2) Github를 활용하여 협업능력 향상
- 3) 개인 메신저와 그룹웨어를 하나의 프로젝트로 구현하여 편리성을 가진 시스템 구축

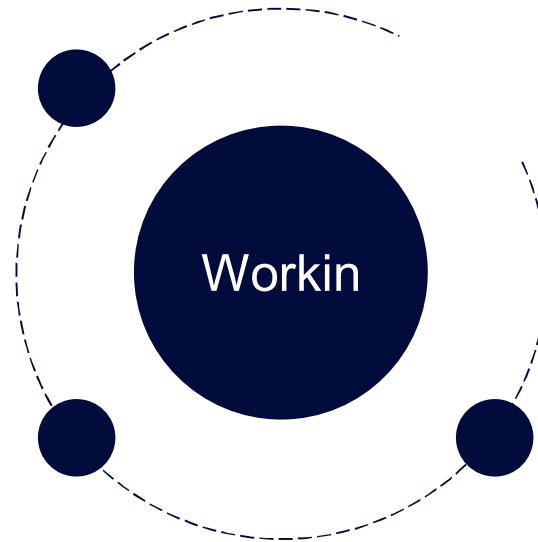
개발 과정 - 역할분담

조현민

로그인 / 회원가입

게시판

클라우드



김태현

전체 회원 일정 테이블

일정 페이지

- 캘린더

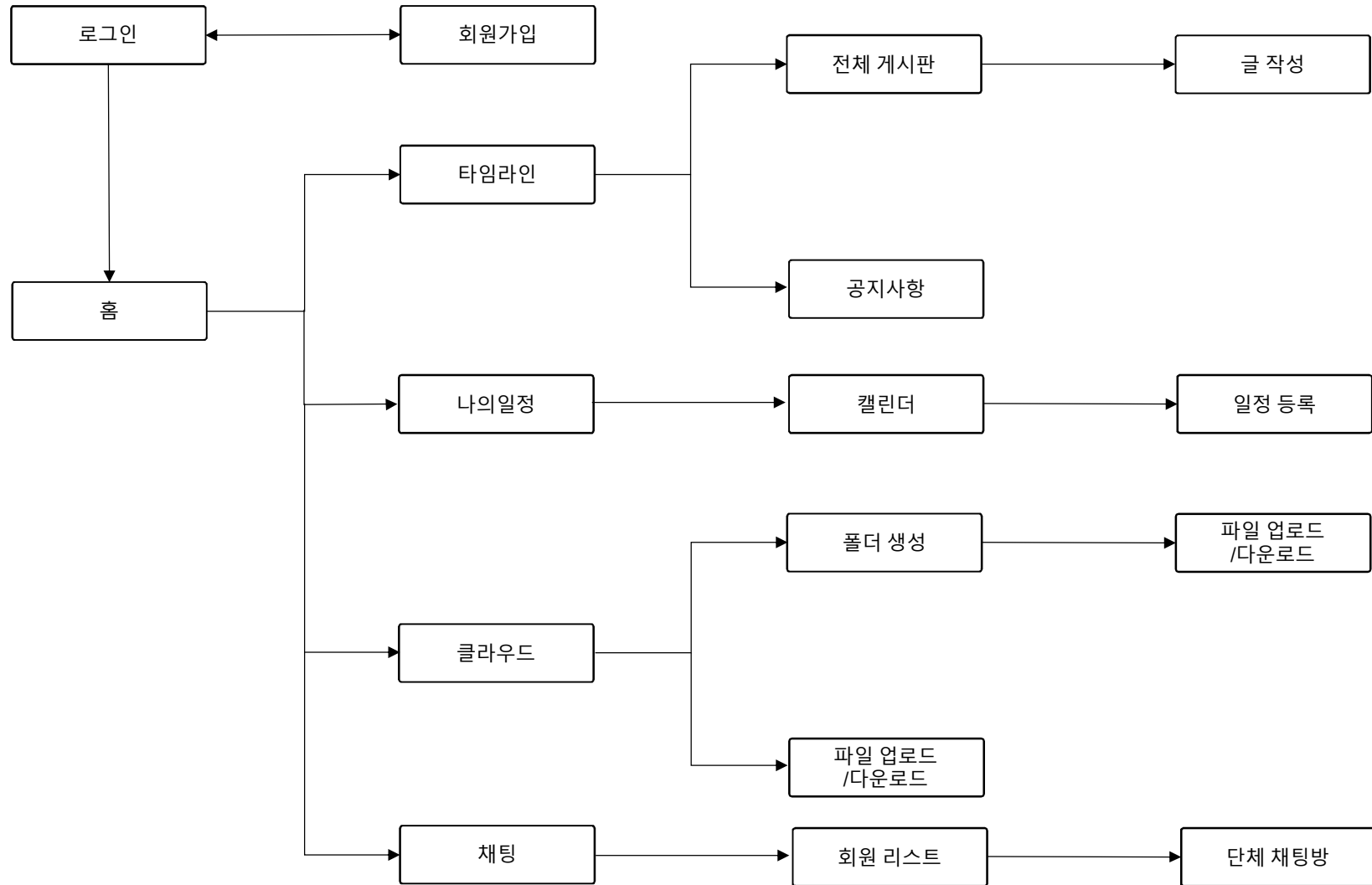
김도균

실시간 채팅 기능

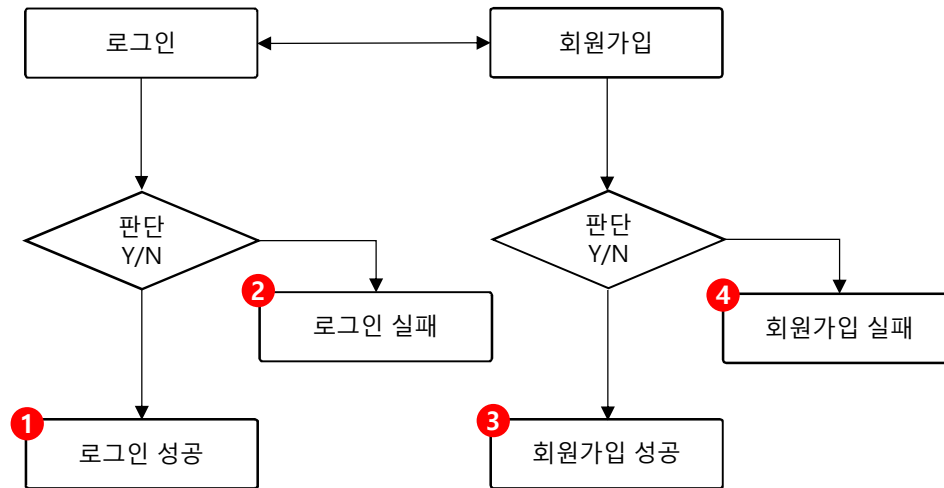
- 실시간 접속자

- 단체 채팅방

플로우차트



로그인 / 회원가입



1. 로그인 성공

- 메인 페이지로 이동
- 채팅 접속

2. 로그인 실패

- 알림창으로 실패 메시지 (정보가 올바르지않습니다)
- 로그인페이지로 이동

3. 회원가입 성공

- 성공 메시지 출력 후 로그인 페이지도 이동
- DB에 회원정보 등록

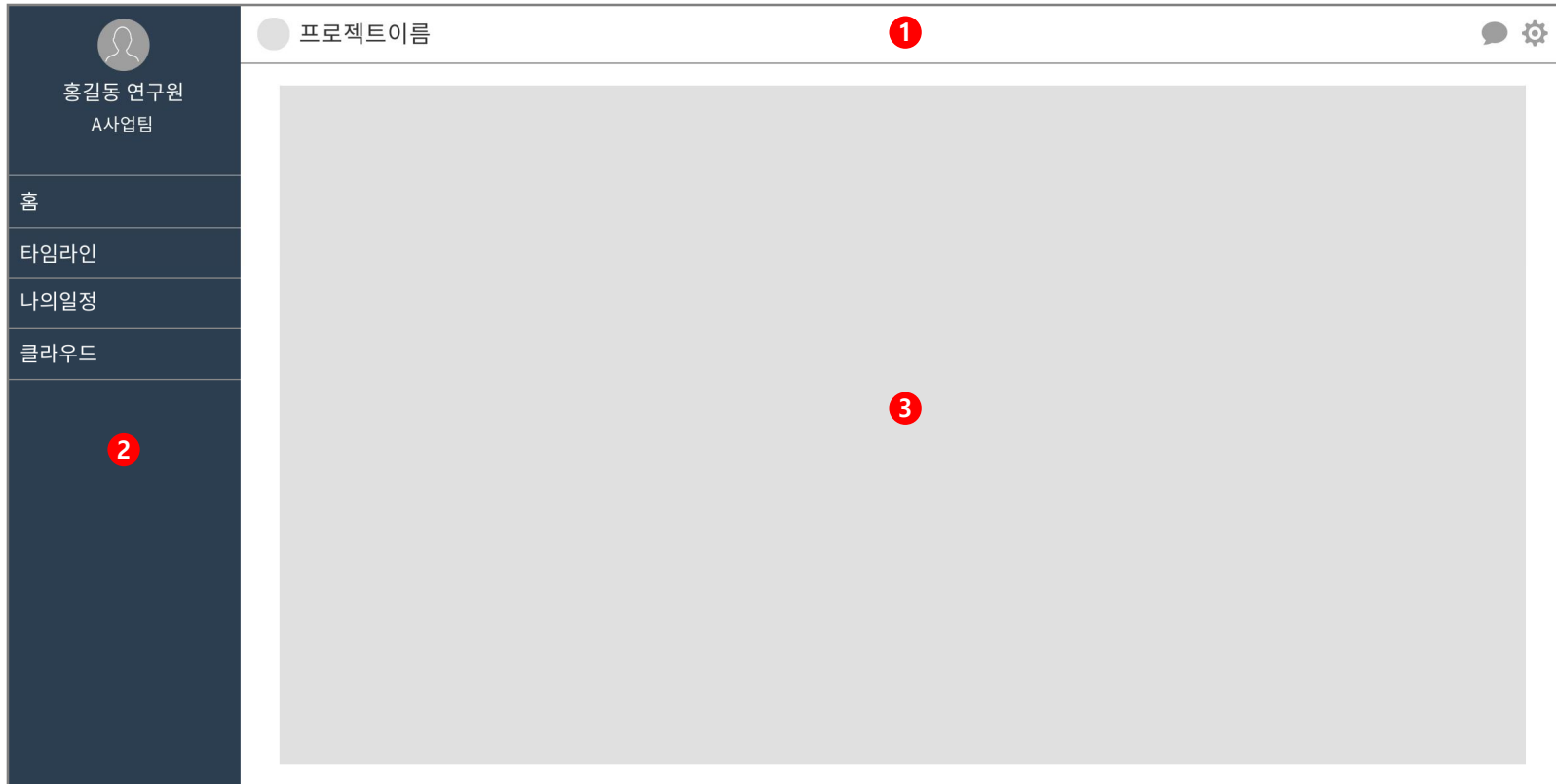
4. 회원가입 실패

- 각각 폼마다 예외처리

레이아웃

전체 프레임 1200 x 720

ovenapp 툴로 제작



1. 상단 메뉴바

- 아이콘 / 프로젝트이름
- 채팅
- 설정

2. 사이드바

- 사이드탭을 이용해 메뉴끼리 이동이 용이하도록

3. 콘텐츠 영역

홈

전체 회원의 주간일정

사용자가 로그인 성공 시 처음 보여지는 화면으로 모든 회원들의 주간 일정 확인이 가능



태현님
A사업팀

홈

타임라인

나의일정

클라우드

Workin

채팅

설정

1

<<

2

2021-06<>>

4

	6 / 일	7 / 월	8 / 화	9 / 수	10 / 목	311 / 금 [today]	12 / 토
태현		외근 PM 4				회의 AM 9	
현민		외근 AM 11		회의 PM 5		회의 PM 3	
도균			외근 PM 1	재택근무			
고고					회의 PM 6		
나나		재택근무	재택근무				
다다				휴가	휴가	휴가	
라라		회의 AM 11	외근 PM 3	외근 PM 4	회의 PM 2		
루이		재택근무	재택근무	재택근무	회의 PM 3	회의 PM 3	
하나		회의 AM 10		회의 PM 3		회의 AM 9	

1. 날짜 이동 버튼

좌 우 두개의 버튼을 이용하여
주간 단위로 이동 가능

2. 날짜 라벨

클릭시 이번주로 돌아옴

3. today

오늘 날짜를 표기

4. 주간 테이블

모든 회원의 주간 일정을
테이블로 구현

홈 - 주간 테이블

1) 오늘 날짜가 포함된 주의 날짜 배열 얻어오기

```
//현재 한 주의 값 구하기
public void setDate() {
    int [] array;
    int yy = currentDate.get(Calendar.YEAR);
    int mm = currentDate.get(Calendar.MONTH);
    String m = getZeroString(mm+1);
    String ym = Integer.toString(yy) + m;
    int wm = currentDate.get(Calendar.WEEK_OF_MONTH);
    array = getRangeDateOfWeek(ym, wm);

    for(int i=0;i<array.length;i++) {
        if(i==0) {
            dates[i+1] = Integer.toString(array[i])+" / 일";
        }else if(i==1) {
            dates[i+1] = Integer.toString(array[i])+" / 월";
        }else if(i==2) {
            dates[i+1] = Integer.toString(array[i])+" / 화";
        }else if(i==3) {
            dates[i+1] = Integer.toString(array[i])+" / 수";
        }else if(i==4) {
            dates[i+1] = Integer.toString(array[i])+" / 목";
        }else if(i==5) {
            dates[i+1] = Integer.toString(array[i])+" / 금";
        }else if(i==6) {
            dates[i+1] = Integer.toString(array[i])+" / 토";
        }
    }
    showToday();
}
```

한 주의 값을 구해서 array배열에 저장
array에는 일 ~ 토요일 순서대로 날짜가 저장됨

```
//오늘 표시하기
public void showToday() {
    int yy = currentDate.get(Calendar.YEAR);
    int mm = currentDate.get(Calendar.MONTH);
    String d = null;
    for(int i=0;i<dates.length-1;i++) {
        if(dates[i+1].length()==5) {
            d = dates[i+1].substring(0, 1);
        }else if(dates[i+1].length()==6) {
            d = dates[i+1].substring(0, 2);
        }
        int da = Integer.parseInt(d);
        if(da==dd_t && yy==yy_t && mm==mm_t ) {
            dates[i+1] = dates[i+1]+ " [today]";
        }
    }
}
```

현재의 날짜와 오늘을 비교해서
일치하다면 today를 추가하여 오늘임을
표시 해줌

홈 - 주간 테이블

2) 지난 주 배열 구하기

```
public void setDateFirst() {  
    int yy = currentDate.get(Calendar.YEAR);  
    int mm = currentDate.get(Calendar.MONTH);  
    int dd = currentDate.get(Calendar.DATE);  
  
    if(dayOfWeek(Integer.toString(yy),Integer.toString(mm+1),Integer.toString(dd)) == 0) { //일요일  
        currentDate.set(yy, mm, dd);  
    } else if(dayOfWeek(Integer.toString(yy),Integer.toString(mm+1),Integer.toString(dd)) == 1) { //월요일  
        currentDate.set(yy, mm, dd-1);  
    } else if(dayOfWeek(Integer.toString(yy),Integer.toString(mm+1),Integer.toString(dd)) == 2) { //화요일  
        currentDate.set(yy, mm, dd-2);  
    } else if(dayOfWeek(Integer.toString(yy),Integer.toString(mm+1),Integer.toString(dd)) == 3) { //수요일  
        currentDate.set(yy, mm, dd-3);  
    } else if(dayOfWeek(Integer.toString(yy),Integer.toString(mm+1),Integer.toString(dd)) == 4) { //목요일  
        currentDate.set(yy, mm, dd-4);  
    } else if(dayOfWeek(Integer.toString(yy),Integer.toString(mm+1),Integer.toString(dd)) == 5) { //금요일  
        currentDate.set(yy, mm, dd-5);  
    } else if(dayOfWeek(Integer.toString(yy),Integer.toString(mm+1),Integer.toString(dd)) == 6) { //토요일  
        currentDate.set(yy, mm, dd-6);  
    }  
}
```

이전 버튼을 클릭할 경우 현재 요일에 따라서 dd의 값을 다르게 설정

모두 해당 주의 일요일을 가리키도록 설정

해당 주의 값을 테이블의 헤더에 반영시키고
currentDate의 date위치의 값을 -7을 해주어
이전 주로 넘어가도록 한다

```
//저번 주 구하기  
public void prev_week() {  
    TableColumnModel tcm = table_header.getColumnModel();  
    TableColumn tc ;  
  
    int [] array;  
    int yy = currentDate.get(Calendar.YEAR);  
    int mm = currentDate.get(Calendar.MONTH);  
    int dd = currentDate.get(Calendar.DATE);  
  
    String m = getZeroString(mm+1);  
    String ym = Integer.toString(yy) + m;  
    int wm = currentDate.get(Calendar.WEEK_OF_MONTH);  
    array = getRangeDateOfWeek(ym, wm-1);  
    for(int i=0;i<array.length;i++) {  
        if(i==0) {  
            dates[i+1] = Integer.toString(array[i]) + " / 일";  
        } else if(i==1) {  
            dates[i+1] = Integer.toString(array[i]) + " / 월";  
        } else if(i==2) {  
            dates[i+1] = Integer.toString(array[i]) + " / 화";  
        } else if(i==3) {  
            dates[i+1] = Integer.toString(array[i]) + " / 수";  
        } else if(i==4) {  
            dates[i+1] = Integer.toString(array[i]) + " / 목";  
        } else if(i==5) {  
            dates[i+1] = Integer.toString(array[i]) + " / 금";  
        } else if(i==6) {  
            dates[i+1] = Integer.toString(array[i]) + " / 토";  
        }  
    }  
    showToday();  
    for(int i=0;i<dates.length;i++) {  
        tc = tcm.getColumnModel(i);  
        tc.setHeaderValue(dates[i]);  
        System.out.println(dates[i]);  
        table_header.repaint();  
    }  
    currentDate.set(yy, mm, dd-7);  
}
```

홈 - 회원의 이름과 일정

1) 회원의 이름 구하고 테이블에 출력

```
//멤버 이름과 아이디 가져오기
public void getMName(){
    String sql = "select member_id , user_name from member order by member_id asc";
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    try {
        pstmt = con.prepareStatement(sql);
        rs = pstmt.executeQuery();

        rs.last();
        int total = rs.getRow();
        record = new String[total][8];
        nm = new Integer[total];
        rs.beforeFirst();
        int index = 0;

        while(rs.next()) {
            record[index][0] = rs.getString("user_name");
            nm[index] = rs.getInt("member_id");
            index++;
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        release(pstmt, rs);
    }
}
```

멤버의 이름을 오름차순으로 구함

테이블의 값이 들어가는 record배열의 크기를
멤버수 만큼으로 지정

record의 [index][0]번째에 회원의 이름을 입력

홈 - 회원의 이름과 일정

2) 테이블에 회원들의 일정 출력

```
for(int i=0;i<c_member.length;i++) {  
    String sql = "select c.member_id, c.cal_category, c.cal_writer, c.cal_date, c.cal_time, c.year, c.month, c.date";  
    sql += " from calendar as c join member as m";  
    sql += " on c.member_id = m.member_id";  
    sql += " where c.member_id = "+nm[i];  
  
    try {  
        pstmt = con.prepareStatement(sql);  
        rs = pstmt.executeQuery();  
        rs.beforeFirst();  
        while(rs.next()) {  
            member_id = rs.getInt("member_id");  
            year = rs.getInt("year");  
            month = rs.getInt("month");  
            date = rs.getInt("date");  
  
            n_category = rs.getInt("cal_category");  
            cal_date = rs.getString("cal_date");  
            cal_time = rs.getString("cal_time");  
  
            if(year == yy && month == (mm+1)) {  
                for(int k=0;k<array.length;k++) {  
                    if(date == array[k]) {  
                        String v = Integer.toString(year);  
                        String m1 = Integer.toString(month);  
                        String d = Integer.toString(date);  
                        w = dayOfWeek(v, m1, d); // 해당 일의 요일의 값을 반환  
                        if(record[i][0] == c_member[i]) {  
                            if(n_category==1) {  
                                record[i][w+1] = "회의 "+cal_date+" "+ cal_time;  
                            }else if(n_category==2) {  
                                record[i][w+1] = "외근 "+cal_date+" "+ cal_time;  
                            }else if(n_category==3) {  
                                record[i][w+1] = "재택근무";  
                            }else if(n_category==4) {  
                                record[i][w+1] = "휴가";  
                            }  
                        }  
                    }  
                }  
            }  
        }  
    }  
} catch (SQLException e) {  
    e.printStackTrace();  
} finally {  
    release(pstmt, rs);  
}
```

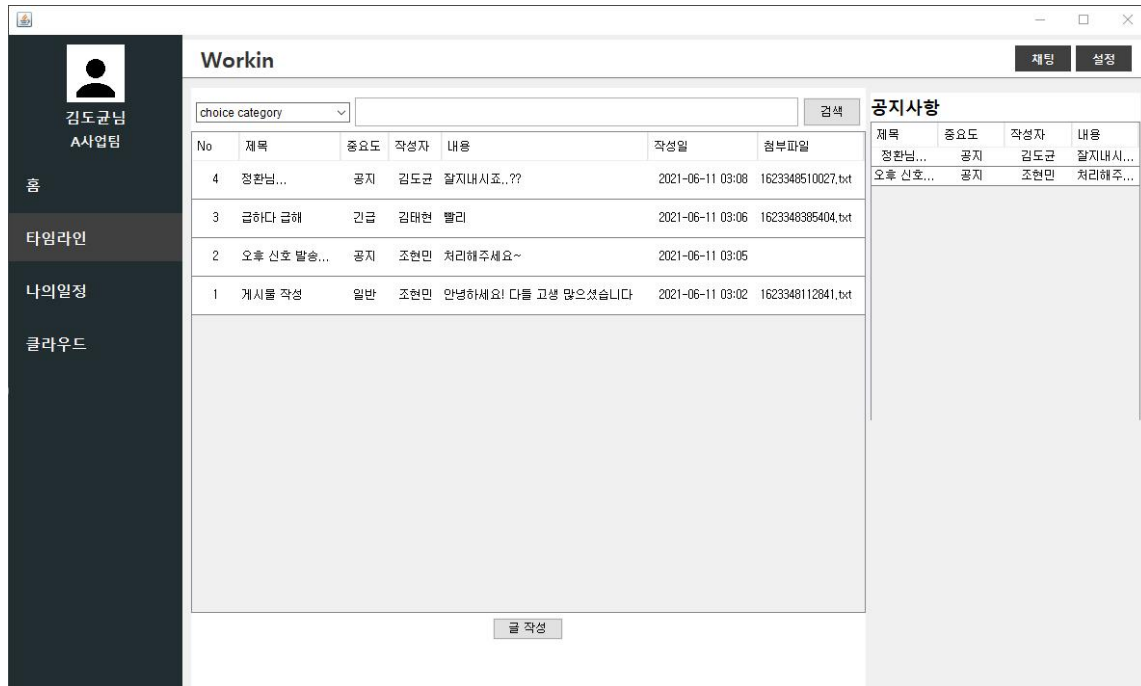
캘린더 테이블에 저장된 일정의 year와 month값을 비교하고

한주의 값이 담긴 array의 길이만큼 반복하면서 array의 k번째에 date의 값이 있는지 파악한다.

일정의 category값을 비교하여 “회의” 또는 “외근” 일 경우 시간도 함께 출력해준다.

타임라인(게시판)

화면 및 주요기능



글작성

- 일반, 중요, 긴급 3가지 분류로 업무의 중요도를 색상과 카테고리로 구분
- 글작성 시 파일첨부 가능
- 최신작성글이 화면 상단부터 위치
- 스크롤 혹은 페이지네이션

검색 기능

- 작성자 이름 / 제목의 키워드 2가지 항목을 탭에서 선택후 검색 가능

타임라인(게시판)

테이블 생성 및 DB 정보 출력

```
public void getNoticeList() {
    PreparedStatement pstmt = null;
    ResultSet rs = null;

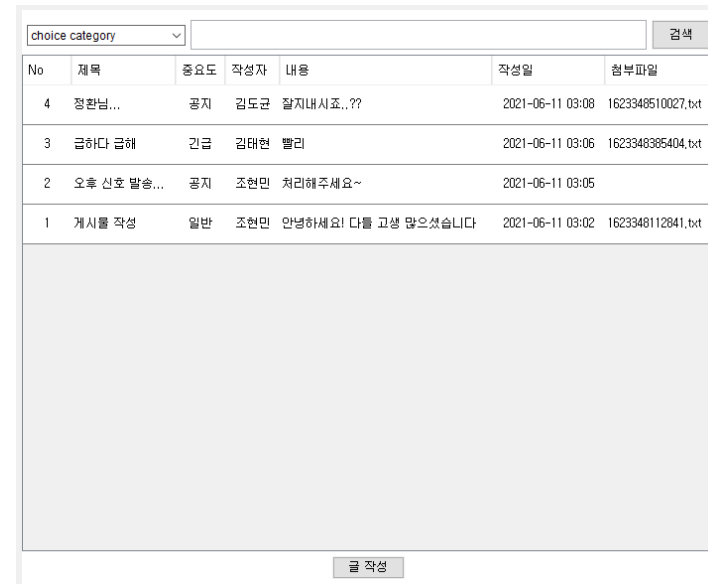
    String sql = "select notice_id, title, level, writer, content, regdate, filename from notice";
    sql += " order by notice_id desc";

    try {
        pstmt=this.getAppMain().getMemberMain().getCon().prepareStatement(sql
            , ResultSet.TYPE_SCROLL_INSENSITIVE
            , ResultSet.CONCUR_READ_ONLY);

        rs = pstmt.executeQuery();
        rs.last();
        int total = rs.getRow();

        //JTable이 참조하고 있는 records이차원배열의 값을, JTable을 이용하여 갱신
        records=new String[total][columns.length];

        rs.beforeFirst(); // 커서를 제자리로
        int index = 0;
        while(rs.next()) {
            date = rs.getTimestamp("regdate");
            dateFormat =new SimpleDateFormat("yyyy-MM-dd HH:mm");
            regdate = dateFormat.format(date);
            records[index][0]=rs.getString("notice_id");
            records[index][1]=rs.getString("title");
            records[index][2]=rs.getString("level");
            records[index][3]=rs.getString("writer");
            records[index][4]=rs.getString("content");
            records[index][5]=regdate;
            records[index][6]=rs.getString("filename");
            index++;
        }
        table.updateUI();//JTable 갱신
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        this.getAppMain().getMemberMain().release(pstmt, rs);
    }
}
```



No	제목	중요도	작성자	내용	작성일	첨부파일
4	정환님...	공지	김도균	잘지내시죠..??	2021-06-11 03:08	1623348510027.txt
3	급하다 급해	긴급	김태현	빨리	2021-06-11 03:06	1623348385404.txt
2	오후 신호 발송...	공지	조현민	처리해주세요~	2021-06-11 03:05	
1	게시물 작성	일반	조현민	안녕하세요! 다들 고생 많으셨습니다	2021-06-11 03:02	1623348112841.txt

1. 쿼리문

내림차순으로 데이터 선택

2. 테이블 갱신

JTable이 참조하고 있는 이차원배열의 값을
ResultSet을 이용하여 갱신

타임라인(게시판)

게시글 작성

신규 게시물 작성

일반

게시물 작성

안녕하세요! 다들 고생 많으셨습니다

첨부파일 : -----파일 첨부등록취소

신규 게시물 작성

일반

게시물 작성

안녕하세요! 다들 고생 많으셨습니다

Message

알림 등록 완료

OK

첨부파일 : Test1.txt파일 첨부등록취소

choice category

검색

No	제목	중요도	작성자	내용	작성일	첨부파일
1	게시물 작성	일반	조현민	안녕하세요! 다들 고생 많으셨습니다	2021-06-11 03:02	1623348112841.txt

1. 글작성 버튼 클릭
입력폼

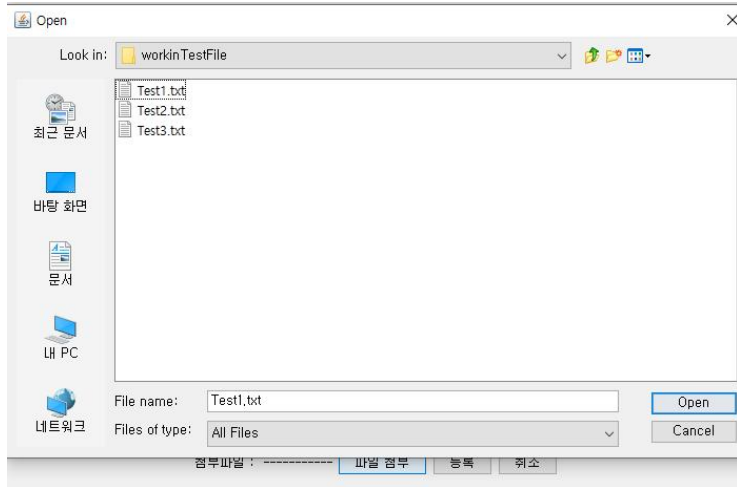
2. 등록
게시글 작성 후 등록

3. 테이블 출력

타임라인(게시판)

게시글 작성 - 파일첨부

실행화면



```
if(chooser.showOpenDialog(this)==JFileChooser.APPROVE_OPTION) {  
    File file = chooser.getSelectedFile();  
    String text = file.getName();  
    try {  
        fis = new FileInputStream(file);  
        long time = System.currentTimeMillis();  
        filename = time+"."+FileManager.getExtend(file.getAbsolutePath(),"\\");  
        fos = new FileOutputStream("D:\\Korea202102_javaworkspace\\Final\\data"+filename);  
  
        //파일 복사  
        int data = -1;  
        byte[] buff = new byte[1024];  
        while(true) {  
            data = fis.read(buff);  
            if(data == -1)break;  
            fos.write(buff);  
        }  
    }  
}
```

JFileChooser로 파일을 선택하고

추후 첨부파일을 다운로드했을 경우 파일 복사해서 가져오기 위해

지정한 폴더에 파일을 복사해 넣기

타임라인(게시판)

게시글 세부정보

실행화면

제목	게시글 작성	중요도	일반
작성자	조현민		
안녕하세요! 다들 고생 많으셨습니다			
첨부파일 : 1623348112841.txt			
<div>파일 다운로드</div> <div>수정</div> <div>삭제</div> <div>취소</div>			

테이블에 있는 게시글 한건 클릭시 보여지는 세부정보
수정 삭제 첨부파일에 들어있는 파일 다운로드 가능

```
public void getData() {  
    String sql = "select title, level, writer, content, filename from notice where notice_id=" + notice_id;  
  
    PreparedStatement pstmt = null;  
    ResultSet rs = null;  
  
    try {  
        pstmt = con.prepareStatement(sql);  
        rs = pstmt.executeQuery();  
  
        if(rs.next()) {  
            t_title.setText(rs.getString("title"));  
            t_level.setText(rs.getString("level"));  
            t_writer.setText(rs.getString("writer"));  
            t_area.setText(rs.getString("content"));  
            la_file.setText(rs.getString("filename"));  
            filename = rs.getString("filename");  
        }  
    }  
}
```

게시글 한건 클릭시 getDate() 메서드가 호출되며 해당 게시글이
보유한 notice_id를 이용하여 쿼리문 실행

나의 일정 페이지

개인 일정 등록 캘린더

버튼을 클릭하면 오늘이 포함된 월로 돌아옴



태현님
A사업팀

홈

타임라인

나의일정

클라우드

Workin

채팅

설정

전년도와 이전 월로 이동

Today 2021/06/11

다음년도와 다음월로 이동

<< <

2021-06


> >>

SUN	MON	TUE	WED	THU	FRI	SAT
		1	2	3	4	5
6	7 외근 PM4	8	9	10	11 회의 AM9	12
13	14	15 재택근무	16	17 회의 PM1	18	19
20	21 휴가	22	23	24	25	26
27	28	29 외근 AM11	30	카테고리별로 색깔을 다르게 지정하여 사용자가 알아보기 쉽도록 함		

오늘을 나타냄

나의 일정 페이지

월별 년도별 캘린더 이동



대원님
A사업팀

홈

타임라인

나의일정


클라우드

Workin

Today 2021/06/11

<< < 2021-05 > >>

SUN	MON	TUE	WED	THU	FRI	SAT
						1
2	3	4	5 재택근무	6	7	8
9	10	11	12	13	14	15
16	17 회의 PM 6	18	19	20	21	22
23	24	25	26	27	28	29
30	31					



대원님
A사업팀

홈

타임라인

나의일정


클라우드

Workin

Today 2021/06/11

<< < 2020-06 > >>

SUN	MON	TUE	WED	THU	FRI	SAT
	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				



대원님
A사업팀

홈

타임라인

나의일정


클라우드

Workin

Today 2021/06/11

<< < 2021-07 > >>

SUN	MON	TUE	WED	THU	FRI	SAT
				1	2	3
4	5	6	7	8 휴가	9 휴가	10
11	12 휴가	13 휴가	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31



대원님
A사업팀

홈

타임라인

나의일정

클라우드

Workin

Today 2021/06/11

<< < 2022-06 > >>

SUN	MON	TUE	WED	THU	FRI	SAT
			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		

나의 일정 페이지

요일을 표현하는 DayBox와 날짜를 표현하는 DateBox로 구성

```
//요일패널 생성
public void createDay() {
    DayBox dayBox;
    for(int i=0;i<7;i++) {
        if(i==0) {
            dayBox=new DayBox(dayArray[i],Color.white,130,23,Color.red);
        }else if(i==6) {
            dayBox=new DayBox(dayArray[i],Color.white,130,23,Color.blue);
        }else {
            dayBox=new DayBox(dayArray[i],Color.white,130,28,Color.black);
        }
        dayBox.setFont(new Font("Arial-Black",100,18));
        p_center.add(dayBox); //센터에 부착!!
    }
}
```

토요일 일요일 평일을 구분하여 요일의 색상을 다르게 설정함

날짜패널은 해당 월의 처음 시작일부터 마지막일까지만 나타나도록 함
- 불필요한 패널들의 색상을 투명으로 설정하여 보이지 않도록 함

```
//파라미터 생성
public void createDate(AppMain appMain) {
    int yy = currentDate.get(Calendar.YEAR);
    int mm = currentDate.get(Calendar.MONTH);
    int firstday = getFirstDayOfMonth(yy, mm);
    DateBox dateBox;

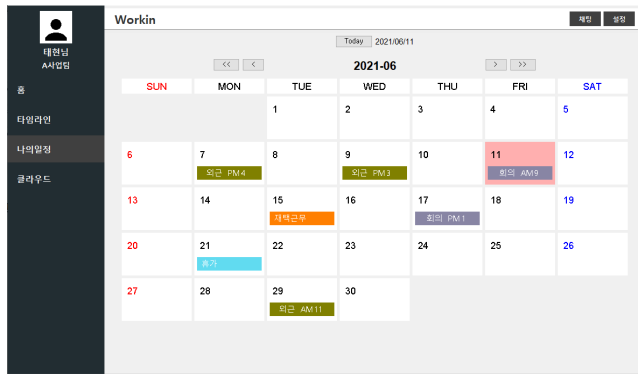
    for(int i=0;i<dayArray.length*6;i++) {
        if(i<firstday|| i-(firstday-1)>getLastDate(yy, (mm+1))) {
            dateBox = new DateBox("", new Color(255,0,0,0), 130,79,Color.black,yy,mm,"", "", "",category,new Color(255,0,0,0)),
            dateBox.click(appMain);
        } else if(i==6 || i==13 || i==20 || i==27) {
            dateBox = new DateBox("", Color.pink, 130,79,Color.black,yy,mm,"", "", "",category,new Color(255,0,0,0),this);
            dateBox.click(appMain);
        } else if(i==0 || i==7 || i==14 || i==21 || i==28){
            dateBox = new DateBox("", Color.white, 130,79,Color.red,yy,mm,"", "", "",category,new Color(255,0,0,0),this);
            dateBox.click(appMain);
        } else if(i==6 || i==13 || i==20 || i==27){
            dateBox = new DateBox("", Color.white, 130,79,Color.blue,yy,mm,"", "", "",category,new Color(255,0,0,0),this);
            dateBox.click(appMain);
        } else {
            dateBox = new DateBox("", Color.white, 130,79,Color.black,yy,mm,"", "", "",category,new Color(255,0,0,0),this);
            dateBox.click(appMain);
        }
        p_center.add(dateBox);
        boxArray[i]=dateBox;
    }
}
```

날짜도 요일을 구분하여 색상을 설정함

오늘날짜의 dateBox패널은 배경색을 pink로 설정함

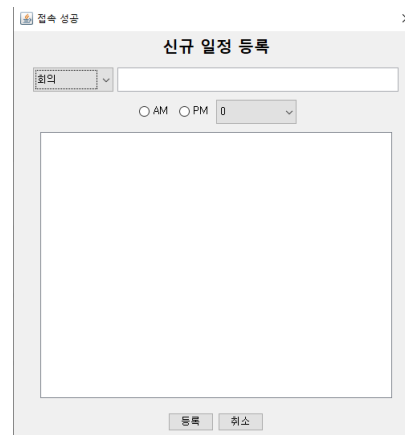
나의 일정 페이지

일정등록



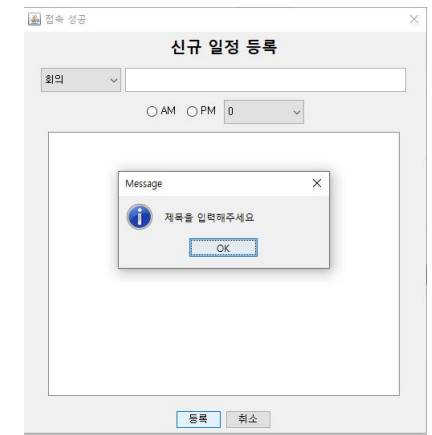
사용자가 일정을 등록하고 싶은 날짜를 클릭

6월 10일 선택함



일정 입력 폼 생성

1. 사용자는 카테고리를 선택 (회의, 외근, 재택근무, 휴가)
2. 제목 작성
3. 시간 작성
4. 내용 작성



사용자가 제목과 시간 내용을 작성하지 않으면
등록할 수 없음

나의 일정 페이지

일정등록

접속 성공

신규 일정 등록

외근

☐ AM ☒ PM 3

강남역에서 만나기로 함

등록 취소



Workin

Today 2021/06/11

2021-06

SUN	MON	TUE	WED	THU	FRI	SAT
		1	2	3	4	5
6	7 외근 PM 4	8	9 외근 PM 3	10 외근 PM 3	11 회의 AM 9	12
13	14	15 재택근무	16	17 회의 PM 1	18	19
20	21 휴가	22	23	24	25	26
27	28	29 외근 AM 11	30			

등록 후에 일정이 캘린더에 나타나는 것을 확인할 수있음

나의 일정 페이지

일정등록

```
public void click(AppMain appMain) {  
    this.addMouseListener(new MouseAdapter() {  
        public void mouseReleased(MouseEvent e) {  
            JPanel p = (JPanel)e.getSource();  
            p.setBackground(Color.RED);  
            d=day;  
            sch = new ScheduleForm(appMain, "신규 일정 등록",appMain,d,year,month, p,calendarmain);  
            dt = new DetailForm(appMain,d,year,month,appMain,calendarmain);  
            if(cal_date=="") {  
                sch.setVisible(true);  
            }else {  
                dt.setVisible(true);  
            }  
        }  
    });  
}
```

날짜를 나타내는 DateBox에 마우스 리스너를 이용하여 해당 날짜의 데이터가 비어있다면 입력폼을 데이터가 들어있다면 상세보기 폼을 나타내도록 한다.

나의 일정 페이지

일정등록

```
String sql = "insert into calendar(member_id, cal_category, cal_title, cal_content, cal_writer,cal_date, cal_time,year,month,date)";
sql+=" values("+id+", "+(r+1)+",?,?,?,?,,?,?)";

PreparedStatement pstmt = null;

try {
    pstmt = con.prepareStatement(sql);
    pstmt.setString(1, m_title);
    pstmt.setString(2, m_content);
    pstmt.setString(3, name);
    pstmt.setString(4, cal_date);
    pstmt.setInt(5, t);
    pstmt.setInt(6, year);
    pstmt.setInt(7, (month+1));
    pstmt.setInt(8, date);
    int result = pstmt.executeUpdate();

    if(result==1) {
        JOptionPane.showMessageDialog(this, "등록성공");
    }else {
        JOptionPane.showMessageDialog(this, "등록실패");
    }
}
```

입력폼의 입력값을 받아와서 sql문으로 db에 저장

나의 일정 페이지

일정등록

DB에 저장된 값을 달력에 표현하기

```
public void paint(Graphics g) {  
  
    g.setColor(color);  
    g.fillRect(0, 0, width,height);  
  
    //날짜  
    g.setColor(c_date); //그래픽 객체의 물감색 = 검정  
    g.setFont(new Font("Arial-Black",100,18));  
    g.drawString(day, 10, 30);  
    g.setColor(cc);  
    g.fillRect(5, 45, 120, 25);  
  
    //제목, 시간  
    if(cate==1) {  
        g.setColor(Color.white);  
        g.setFont(new Font("맑은 고딕",Font.PLAIN,15));  
        g.drawString(title, 26, 63);  
        g.drawString(cal_date, 66, 63);  
        g.drawString(cal_time, 91, 63);  
    }else if(cate==2) {  
  
        g.setColor(Color.white);  
        g.setFont(new Font("맑은 고딕",Font.PLAIN,15));  
        g.drawString(title, 22, 63);  
        g.drawString(cal_date, 62, 63);  
        g.drawString(cal_time, 87, 63);  
    }else if(cate==3) {  
  
        g.setColor(Color.white);  
        g.setFont(new Font("맑은 고딕",Font.PLAIN,15));  
        g.drawString(title, 13, 63);  
    }else if(cate==4) {  
        g.setColor(Color.white);  
        g.setFont(new Font("맑은 고딕",Font.PLAIN,15));  
        g.drawString(title,13, 63);  
    }else if(cate == 0) {  
  
    }  
}
```

dateBox의 paint메서드에서 일정의 카테고리별로 사각형의 색상과 사이즈를 지정해줌

```
String sql = "select * from calendar where member_id="+member_id;  
  
try {  
    pstmt = con.prepareStatement(sql);  
    rs = pstmt.executeQuery();  
    while(rs.next()) {  
        int year = rs.getInt("year");  
        int month = rs.getInt("month");  
        date = rs.getString("date");  
        title = rs.getString("cal_title");  
        category = rs.getInt("cal_category");  
        cal_date = rs.getString("cal_date");  
        cal_time = rs.getString("cal_time");  
  
        String sql2 = "select cal_name from calendar_category where cal_category="+category;  
        pstmt2 = con.prepareStatement(sql2);  
        rs2 = pstmt2.executeQuery();  
        while(rs2.next()) {  
            cate = rs2.getString("cal_name");  
        }  
        if(year==yy && month ==(mm+1)) {  
            for(int i=0;i<boxArray.length;i++) {  
                if(category ==1) {  
                    boxArray[Integer.parseInt(date)+(dd-1)].cc = new Color(136,133,164);  
                    boxArray[Integer.parseInt(date)+(dd-1)].title=cate;  
                    boxArray[Integer.parseInt(date)+(dd-1)].cal_date=cal_date;  
                    boxArray[Integer.parseInt(date)+(dd-1)].cal_time=cal_time;  
                    boxArray[Integer.parseInt(date)+(dd-1)].cate=category;  
                    boxArray[Integer.parseInt(date)+(dd-1)].cate=cate;  
                    boxArray[Integer.parseInt(date)+(dd-1)].repaint();  
                }else if(category==2) {  
                    boxArray[Integer.parseInt(date)+(dd-1)].cc = new Color(128,128,0);  
                    boxArray[Integer.parseInt(date)+(dd-1)].title=cate;  
                    boxArray[Integer.parseInt(date)+(dd-1)].cal_date=cal_date;  
                    boxArray[Integer.parseInt(date)+(dd-1)].cal_time=cal_time;  
                    boxArray[Integer.parseInt(date)+(dd-1)].cate=category;  
                    boxArray[Integer.parseInt(date)+(dd-1)].cate=cate;  
                    boxArray[Integer.parseInt(date)+(dd-1)].repaint();  
                }else if(category==3) {  
                    boxArray[Integer.parseInt(date)+(dd-1)].cc = new Color(255,127,0);  
                    boxArray[Integer.parseInt(date)+(dd-1)].title=cate;  
                    boxArray[Integer.parseInt(date)+(dd-1)].cal_date=cal_date;  
                    boxArray[Integer.parseInt(date)+(dd-1)].cal_time=cal_time;  
                    boxArray[Integer.parseInt(date)+(dd-1)].cate=category;  
                    boxArray[Integer.parseInt(date)+(dd-1)].cate=cate;  
                    boxArray[Integer.parseInt(date)+(dd-1)].repaint();  
                }  
            }  
        }  
    }  
}
```

나의 일정 페이지

일정수정

제목 외부일정

날짜 2021 6 10 시간 PM 3

강남역에서 만나기로 함

수정 삭제 완료

해당 날짜를 클릭하면 상세보기 페이지가 나타남
사용자는 해당 페이지에서 수정과 삭제를 할 수 있음

사용자는 일정의 내용과 제목 날짜 시간을 수정가능

제목 외부일정

날짜 2021 6 11 시간 PM 3

강남역에서 만나기로 함

THU	FRI
3	4
10 외근 PM 3	11 회의 AM 9

이미 일정이 등록되어 있는 곳으로 날짜를 수정할 경우

제목 외부일정

날짜 2021 6 11 시간 PM 3

강남역에서 만나기로 함

Message

이미 일정이 등록되어 있습니다.

OK

수정 삭제 완료

경고창이 나타나고 수정은 이루어지지 않는다.

나의 일정 페이지

일정수정

접속 성공

제목

외부일정

날짜

2021 6 14

시간

PM

3

강남역에서 만나기로 함

미팅은 2시간 정도 예상

수정

삭제

완료

9	10	11	12
외근 PM 3		회의 AM 9	외근 PM 3

날짜 변경시 원래 있던 일정은 사라지고 수정된 날짜에 일정이 생성됨

나의 일정 페이지

일정수정

```
String sql = "select * from calendar where year="+this.year+" and month = "+(this.month+1)+" and date="+date;
try {
    pstmt = con.prepareStatement(sql);
    rs = pstmt.executeQuery();

    if(rs.next()) {
        title = rs.getString("cal_title");
        year = rs.getInt("year");
        month = rs.getInt("month");
        date = rs.getInt("date");
        contents = rs.getString("cal_content");
        cal_date = rs.getString("cal_date");
        cal_time = rs.getString("cal_time");

        f_title.setText(title);
        f_year.setText(Integer.toString(year));
        f_month.setText(Integer.toString(month));
        f_date.setText(Integer.toString(date));
        com_time.setSelectedItem(cal_date);
        f_time2.setText(cal_time);
        content.setText(contents);
    }
}
```

해당 날짜에 저장된 일정의 정보들을 먼저 추출함

```
String sql = "select * from calendar where year="+year+" and month = "+month+" and date="+date+" and member_id = "+member_id;
PreparedStatement pstmt=null;

ResultSet rs =null;
try {
    pstmt = con.prepareStatement(sql);
    rs = pstmt.executeQuery();

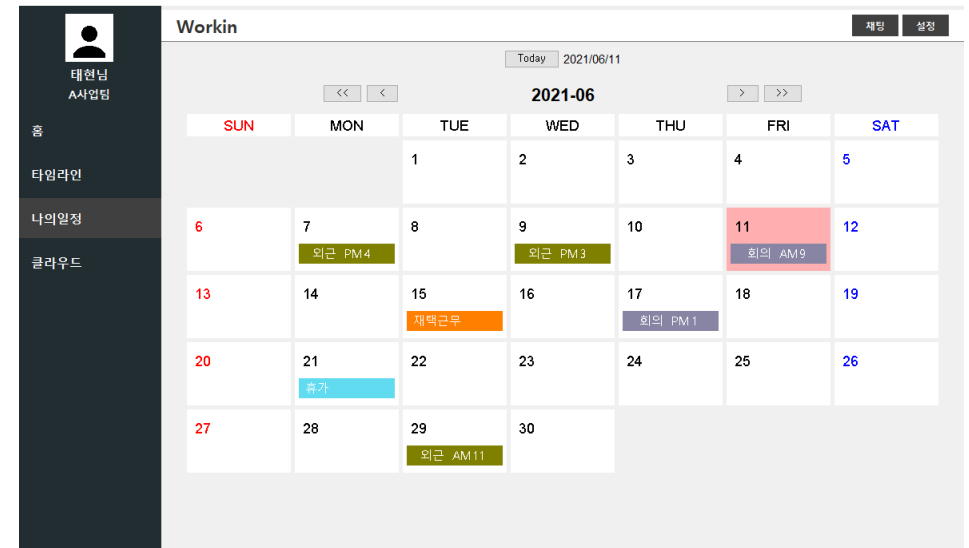
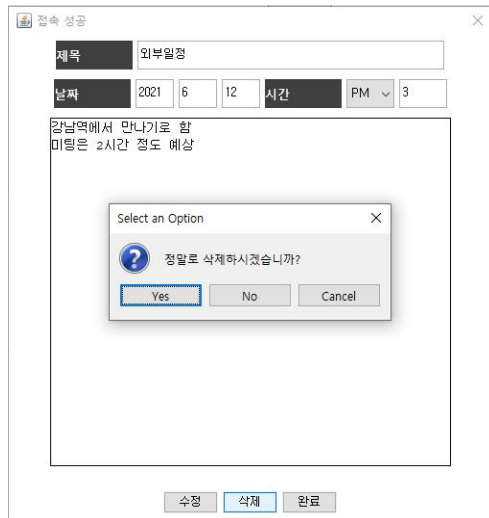
    if(rs.isBeforeFirst()) { //조회된 데이터가 없으면 false 있으면 true - 데이터가 있을 때 실행됨
        if(year == this.year && month == (this.month+1) && date == Integer.parseInt(this.choose)) {
            updateData(appMain);
        }else {
            JOptionPane.showMessageDialog(this, "이미 일정이 등록되어 있습니다.");
        }
    }else{
        updateData(appMain);
    }
} catch (SQLException e) {
```

사용자가 수정한 날짜가 이미 일정이 등록되어 있다면 등록할 수 없도록 함

나의 일정 페이지

일정삭제

사용자가 일정을 삭제하면 캘린더에 나타나있던 일정도 삭제됨



나의 일정 페이지

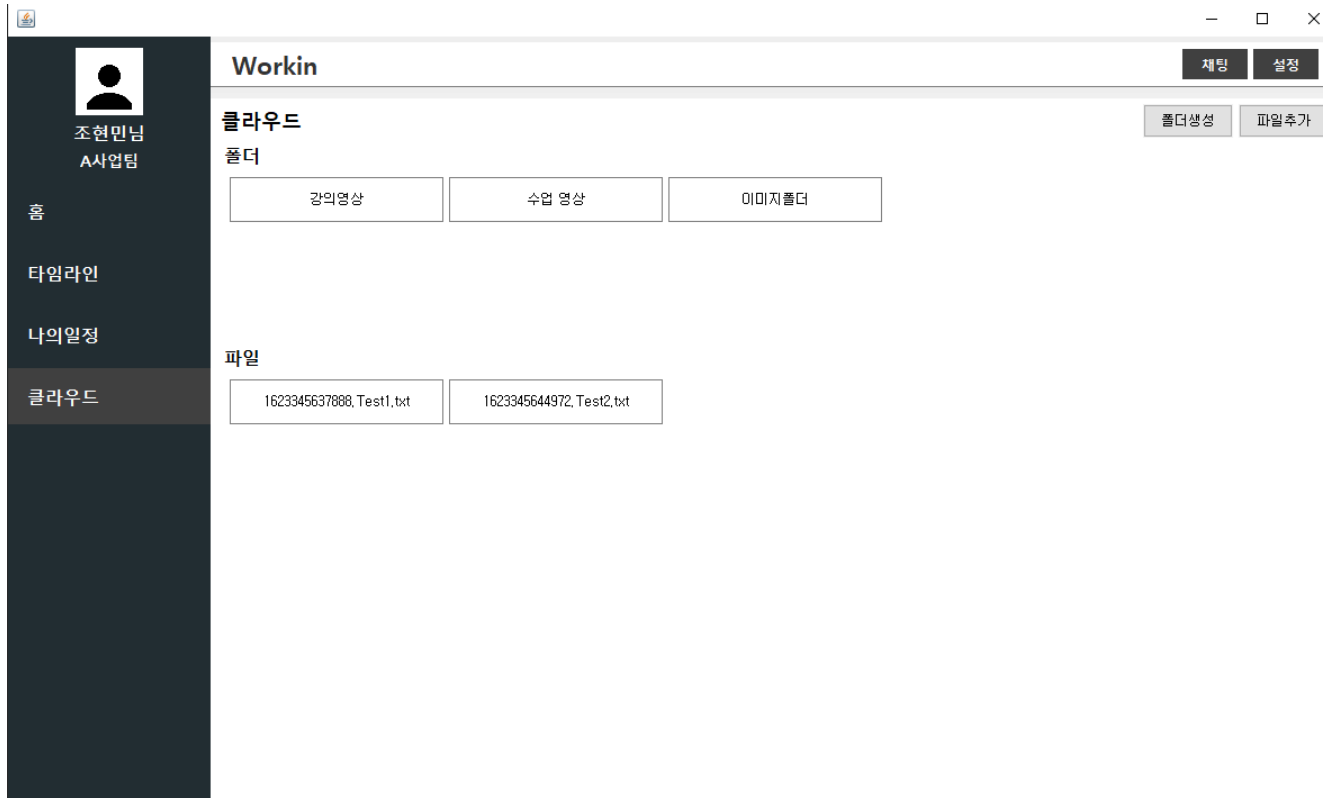
일정수정

```
String sql = "delete from calendar where year="+this.year+" and month = "+(this.month+1)+" and date="+date;
PreparedStatement pstmt =null;
try {
    pstmt = con.prepareStatement(sql);
    int result2 = JOptionPane.showConfirmDialog(this, "정말로 삭제하시겠습니까?");
    if(result2==JOptionPane.YES_OPTION) { //예
        int result = pstmt.executeUpdate();
        if(result==1) {
            JOptionPane.showMessageDialog(this, "삭제성공");
        }else {
            JOptionPane.showMessageDialog(this, "삭제실패");
        }
    }
} catch (SQLException e) {
    e.printStackTrace();
}
```

사용자가 선택한 날짜의 값을 삭제함

클라우드(문서함)

실행화면

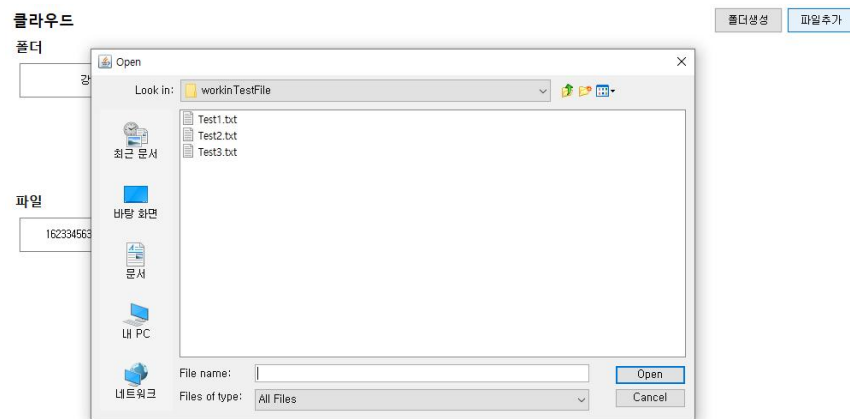


파일을 업로드 다운로드 가능한 페이지 구현
폴더를 생성하여 파일을 관리할 수 있고
폴더없이 파일업로드도 가능하게 구현

클라우드(문서함)

폴더없이 파일 추가하기

실행화면



실행결과

파일

1623345637888, Test1.txt	1623345644972, Test2.txt	1623354769227, Test3.txt
--------------------------	--------------------------	--------------------------

JFileChooser로 파일을 선택해서 등록

클라우드(문서함)

폴더없이 파일 추가하기

```
public void uploadFile() {  
    if(fileCount < 20) {  
        FileInputStream fis = null;  
        FileOutputStream fos = null;  
  
        PreparedStatement pstmt = null;  
        String sql = "insert into file(file_name, writer)";  
        sql += " values(?, ?)";  
  
        if(chooser.showOpenDialog(null) == JFileChooser.APPROVE_OPTION) {  
            File file = chooser.getSelectedFile();  
  
            try {  
                fis = new FileInputStream(file);  
                long time = System.currentTimeMillis();  
                filename = time + "." + file.getName();  
                fos = new FileOutputStream("D:\\Korea202102_javaworkspace\\Final\\cloud"+filename);  
  
                //파일 복사  
                int data = -1;  
                byte[] buff = new byte[1024];  
                while(true) {  
                    data = fis.read(buff);  
                    if(data == -1) break;  
                    fos.write(buff);  
                }  
  
                pstmt = this.getAppMain().getMemberMain().getCon().prepareStatement(sql);  
                pstmt.setString(1, filename);  
                pstmt.setString(2, this.appMain.getMember().getUser_name());  
  
                pstmt.executeUpdate();  
                JOptionPane.showMessageDialog(this, "업로드 완료");  
            }  
        }  
    }  
}
```

1) 파일은 최대 20개까지 업로드 가능하게 구현

2) 추후 파일을 다운로드했을 경우 파일 복사해서 가져오기 위해
지정한 폴더에 파일을 복사해 넣기

파일명은 currentTimeMillis() + 파일명 + 확장자

파일

1623345637888, Test1.txt

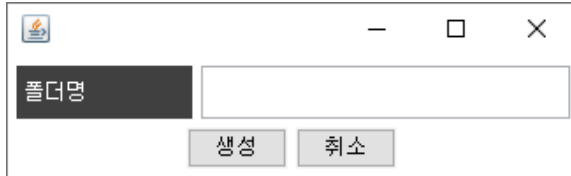
1623345644972, Test2.txt

1623354769227, Test3.txt

3) 변환된 파일명으로 데이터베이스에 추가

클라우드(문서함)

폴더 추가하기



A small window titled "폴더명" (Folder Name) with a text input field and two buttons: "생성" (Create) and "취소" (Cancel).

```
public void createFolder() {  
    String folder_name = folderForm.t_title.getText();  
    PreparedStatement pstmt = null;  
    String sql = "insert into folder(folder_name)";  
    sql += " values(?)";  
  
    try {  
        pstmt = this.getAppMain().getMemberMain().getCon().prepareStatement(sql);  
        pstmt.setString(1, folder_name);  
        pstmt.executeUpdate();  
        JOptionPane.showMessageDialog(this, "폴더 생성 완료");  
    } catch (SQLException e) {  
        e.printStackTrace();  
    } finally {  
        this.getAppMain().getMemberMain().release(pstmt);  
    }  
}
```

1) 입력폼에 입력한 폴더명을 folder_name 변수에 담기

2) 바인드변수에 folder_name을 넣어주고 쿼리문 실행

폴더

강의영상

수업 영상

이미지폴더

클라우드(문서함)

폴더없이 업로드한 파일 불러오기

```
public void getFileList() {
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    String sql = "select file_name, regdate, writer from file where folder_id is null";
    String writer = appMain.getMember().getUser_name();
    try {
        pstmt = this.getAppMain().getMemberMain().getCon().prepareStatement(sql);

        rs = pstmt.executeQuery();
        rs.last(); //커서를 맨 밑으로
        fileCount = rs.getRow(); // 총개수를 구함
        System.out.println("파일의 갯수는 "+fileCount);
        p_file.removeAll(); //패널의 자식 컴포넌트를 모두 삭제
        rs.beforeFirst();

        for(int i = 0; i < fileCount; i++) {
            customFile = new CustomFile[fileCount];
            if(rs.next()) {
                customFile[i] = new CustomFile(rs.getString("file_name"));
                customFile[i].setFile_name(rs.getString("file_name"));
                customFile[i].setRegdate(rs.getString("regdate"));
                customFile[i].setWriter(writer);
            }
            customFile[i].setId(i);
            customFile[i].setPreferredSize(new Dimension(190,40));
            customFile[i].setFocusPainted(false);
            customFile[i].setContentAreaFilled(false);
            customFile[i].setBorder(BorderFactory.createLineBorder(Color.GRAY));
            customFile[i].addActionListener(this);
            //System.out.println(customFile[i].getId());
            p_file.add(customFile[i]);
        }
        p_file.updateUI(); //UI 업데이트!!
    } catch (SQLException e) {
        e.printStackTrace();
    }
}
```

folder_id	file_id	file_name
NULL	1	1623345637888.Test1.txt
NULL	2	1623345644972.Test2.txt

- 1) 폴더에 업로드된 파일들은 각 폴더의 아이디값을 포린키로 지정하는데 폴더 없이 업로드된 파일은 null값
- 2) 쿼리문 실행 후 반환된 데이터들을 rs객체 담아주고
커서를 맨 밑으로 이동후 총 개수를 구하여 fileCount변수에 보관
다시
- 3) 총개수 만큼 반복문을 통해 파일 배열 생성
생성된 배열의 데이터 하나씩 파일이름과 생성일 작성자
아이디값을 담아주고 클릭이벤트 구현

클라우드(문서함)

폴더에 업로드한 파일 불러오기

```
public void getFolderList() {
    PreparedStatement pstmt = null;
    ResultSet rs = null;

    String sql = "select folder_id, folder_name from folder";

    try {
        pstmt = this.getAppMain().getMemberMain().getCon().prepareStatement(sql);

        rs = pstmt.executeQuery();
        rs.last();
        folderCount = rs.getRow();
        p_folder.removeAll();
        rs.beforeFirst();

        for(int i = 0; i < folderCount; i++) {
            customFolder = new CustomFolder(folderCount);
            if(rs.next()) {
                customFolder[i] = new CustomFolder(rs.getString("folder_name"));
                customFolder[i].setFolder_name(rs.getString("folder_name"));
                customFolder[i].setFolder_id(i+1);
                customFolder[i].setPreferredSize(new Dimension(190,40));
                customFolder[i].setFocusPainted(false);
                customFolder[i].setContentAreaFilled(false);
                customFolder[i].setBorder(BorderFactory.createLineBorder(Color.GRAY));

                customFolder[i].addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                        CustomFolder obj = (CustomFolder)e.getSource();

                        System.out.println(obj.getFolder_id());
                        folderDetail = new FolderDetail(CloudMain.this, obj);
                    }
                });
            }
        }
        p_folder.add(customFolder[i]);
    }
}
```

folder_id	folder_name	regdate
1	강의영상	2021-06-10 02:42:02
2	수업 영상	2021-06-10 03:46:22
3	이미지폴더	2021-06-10 03:48:08

1) folder 테이블에 있는 모든 폴더 선택

2) 쿼리문 실행 후 반환된 데이터들을 rs객체 담아주고

커서를 맨 밑으로 이동후 총 개수를 구하여 folderCount변수에
보관

다시

3) 총개수 만큼 반복문을 통해 폴더 배열 생성

생성된 배열의 데이터 하나씩 폴더이름과 생성일 아이디값을
담아주고 클릭이벤트 구현

클라우드(문서함)

파일 다운로드

파일명	1623345637888_Test1.txt
작성자	조현민
생성일	2021-06-11 02:20:37.0
<input type="button" value="다운로드"/> <input type="button" value="삭제"/>	

```
public void downloadFile() {
    FileInputStream fis = null;
    FileOutputStream fos = null;
    int result = JOptionPane.showConfirmDialog(this, this.getFile_name() + " 다운로드 하시겠습니까?", "다운로드", JOptionPane.YES_NO_OPTION);

    if(result == JOptionPane.YES_OPTION) {
        File file = new File("D:\\Korea202102_javaworkspace\\WorkinApp\\cloud\\"+file_name);
        try {
            fis = new FileInputStream(file);
            fos = new FileOutputStream("C:\\Users\\master\\Desktop\\Workin에서 받은 파일\\"+ file_name);

            //파일 복사
            int data = -1;
            byte[] buff = new byte[1024];

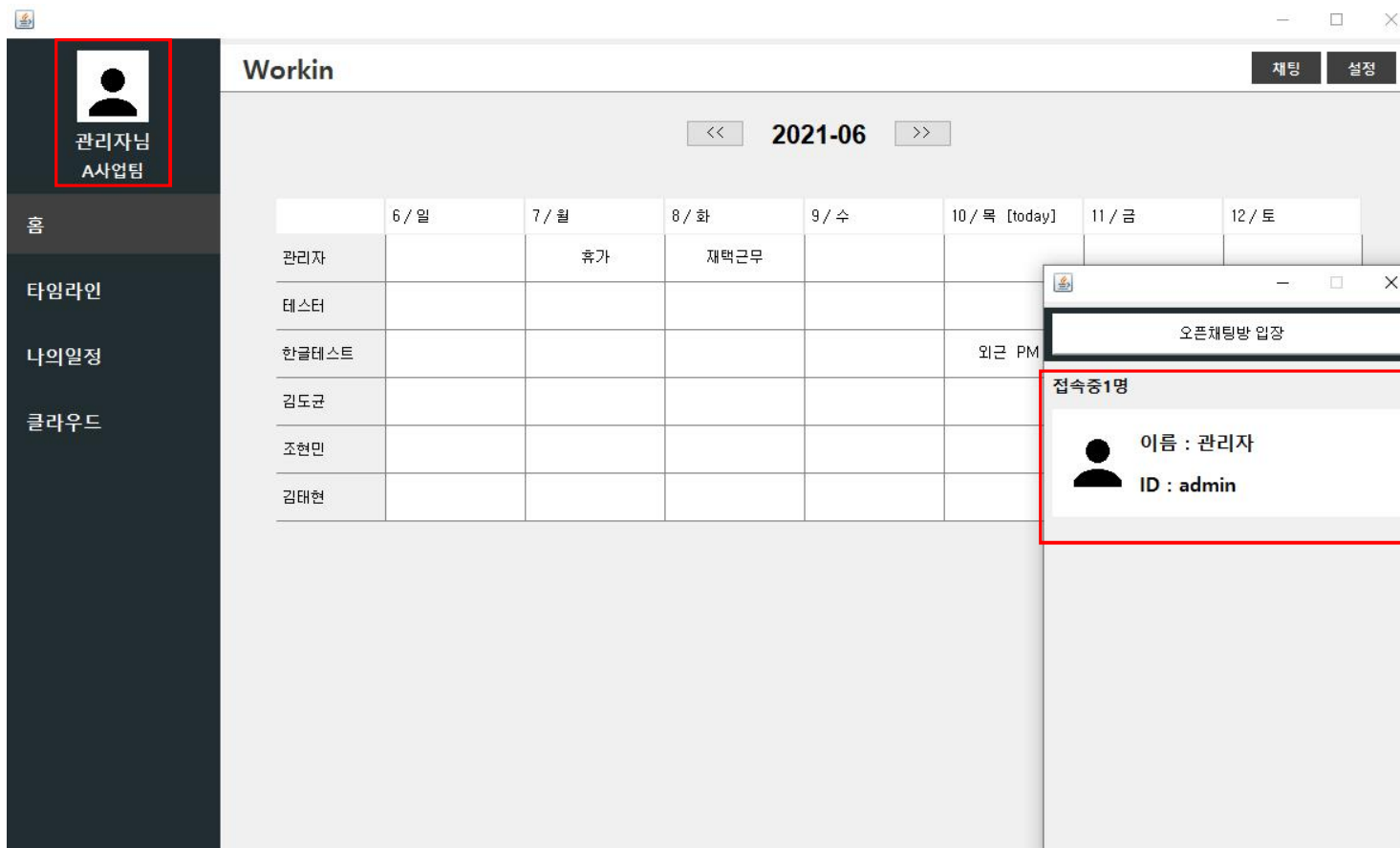
            while(true) {
                data = fis.read(buff);
                if(data == -1)break;
                fos.write(buff);
            }
        }
    }
}
```

- 1) 각 파일마다 담아두었던 file_name과 cloud디렉토리에 복사된 같은 이름을 가진 파일을 지정한 디렉토리로 복사 (마치 다운로드 하는것 처럼 구현)

채팅

클라이언트에서 실시간 접속 나타내기

로그인 성공시 메인화면



The screenshot displays the 'Workin' application interface. On the left is a dark sidebar with navigation links: '홈' (Home), '타임라인' (Timeline), '나의일정' (My Schedule), and '클라우드' (Cloud). The main area shows a calendar for June 2021. A chat window titled '오픈채팅방 입장' (Open Chat Room Entry) is overlaid on the right, showing '접속중1명' (1 user connected) and a user profile for '관리자' (Admin) with ID 'admin'. The user profile is highlighted with a red box.

Workin

채팅 설정

<< 2021-06 >>

	6 / 일	7 / 월	8 / 화	9 / 수	10 / 목 [today]	11 / 금	12 / 토
관리자		휴가	재택근무				
테스터							
한글테스트					외근 PM		
김도균							
조현민							
김태현							

오픈채팅방 입장

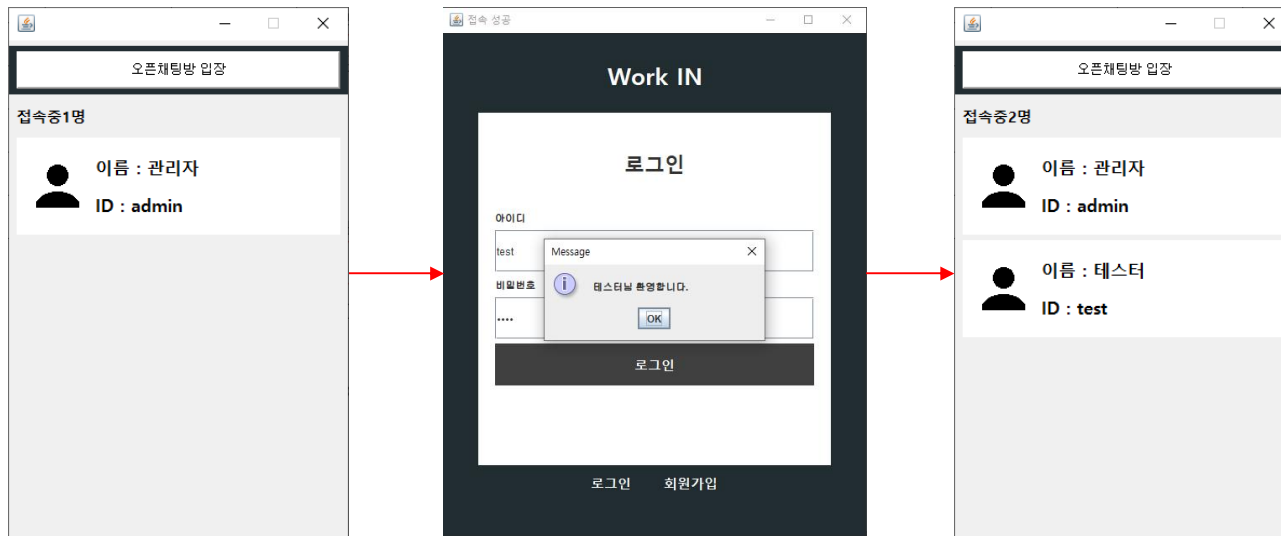
접속중1명

이름 : 관리자
ID : admin

채팅

클라이언트에서 실시간 접속 나타내기

새로운 사람 로그인 시



1. 로그인

메인페이지 호출과 동시에
서버접속

2. 접속 목록

서버측 뿐만 아니라 클라이언트
측에도 현재 로그인 명단 출력

3. 실시간 연동

기존 로그인한 구성원에게도
실시간 브로드캐스팅

채팅

클라이언트에서 실시간 접속 나타내기

```
//보여주기
setBounds(400, 100, 1200, 720);
setVisible(true);
setDefaultCloseOperation(EXIT_ON_CLOSE);

chatready(); //채팅관련 클래스 생성
connect(); //서버와 연결!!!
sendAllData(); // 클라이언트서버 스레드로 로그인 관련정보 전달
}

public void chatready() { //채팅관련 클래스 생성
    chatSelect = new ChatSelect(this);
    chatClient = new ChatClient(this);
}

public void connect() { //서버와 연결!!!
    String ip="192.168.0.3";
    int port=7777;

    try {
        socket = new Socket(ip, port); //접속!!!!!!
        msgThread = new ClientMsgThread(socket, this); //클라이언트 측의 대화용 스레드 생성
        msgThread.start(); // 스레드 run() 실행! 서버의 메시지 실시간 형위 시작
    } catch (UnknownHostException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

public void sendAllData() { //로그인 정보 서버로 전달
    StringBuilder sb = new StringBuilder();
    sb.append("{");
    sb.append("\cmd\" : \"login\","); //cmd : login
    sb.append("\member\":{");
    sb.append("\member_id\" : \"\"+member.getMember_id()+"\",");
    sb.append("\user_name\" : \"\"+member.getUser_name()+"\",");
    sb.append("\user_id\" : \"\"+member.getUser_id()+"\",");
    sb.append("\user_pass\" : \"\"+member.getUser_pass()+"\",");
    sb.append("\regdate\" : \"\"+member.getRegdate()+"\",");
    sb.append("\img\" : \"\"+member.getImg()+"\",");
    sb.append("}");
    sb.append("}");
    msgThread.send(sb.toString()); //서버에 메시지 전송
}
```

1. 로그인 성공
2. 메인 페이지 생성과 동시에 채팅관련클래스 생성
3. ※채팅 클라이언트는 더미 일뿐 실질적 기능은 메인페이지에서 구현됨
4. 서버 연결 및 현재 로그인한 사람의 정보(String)으로 만들어 전달

채팅

클라이언트에서 실시간 접속 나타내기

ClientServerThread

```
public void send(String msg) {  
    try {  
        buffw.write(msg+"\n");  
        buffw.flush();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

cmd:login 정보
(String)전송

```
public void listen() { //끊임없이 듣기 위한 리스너  
    String msg=null;  
    try {  
        msg= buffr.readLine();  
        JSONParser jsonParser=new JSONParser();  
        try {  
            System.out.println("클라이언트 msg 스레드 리스너정보"+msg);  
            JSONObject packet=(JSONObject)jsonParser.parse(msg);  
            String cmd = (String)packet.get("cmd");  
            if(cmd.equals("login")) { //로그인 정보가 전송되어 오면 정보를 VO-Vector에 담아 보내자  
                JSONArray memberArray=(JSONArray)packet.get("memberList");  
                memberList.removeAll(memberList); //누적이 되지않게 기존 데이터 지우기  
  
                for(int i=0;i<memberArray.size();i++) {  
                    member = new Member(); //empty  
                    JSONObject obj=(JSONObject)memberArray.get(i);  
  
                    member.setMember_id(Integer.parseInt((String)obj.get("member_id")));  
                    member.setUser_name((String)obj.get("user_name"));  
                    member.setUser_id((String)obj.get("user_id"));  
                    member.setUser_pass((String)obj.get("user_pass"));  
                    member.setRegdate((String)obj.get("regdate"));  
                    member.setImg((String)obj.get("img"));  
  
                    memberList.add(member);  
                }  
                appMain.savemain(memberList);  
                appMain.chatSelect.onlinecount(memberList);  
            }  
        }  
    }  
}
```

ServerThread

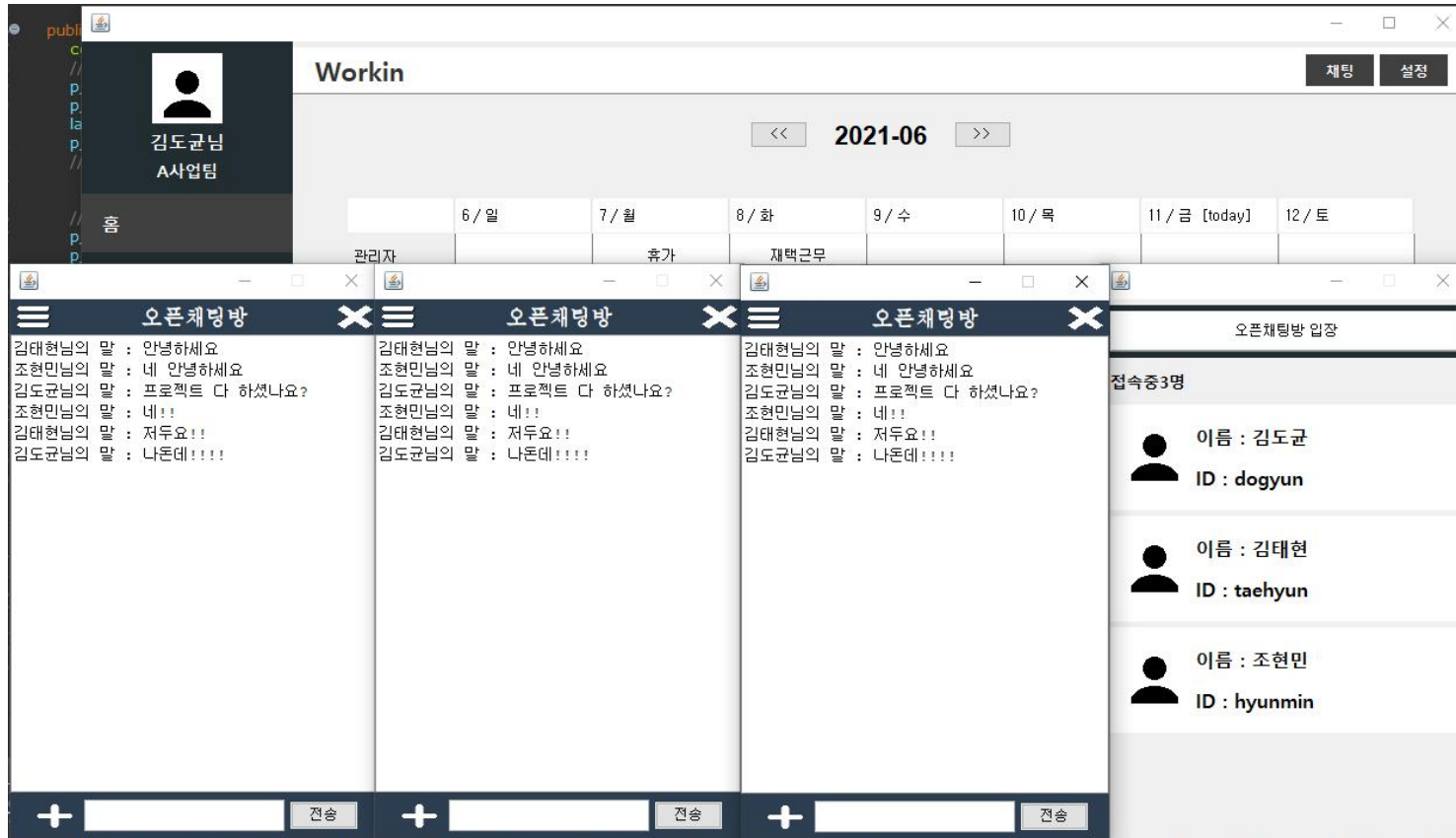
```
public void listen() {  
    String msg=null;  
    try {  
        msg= buffr.readLine(); // 클라이언트가 보낸 msg 이 포함된 스트링발의(=sb(String))를 받아와 파싱을 한다  
        //JSON 파싱 시작!!!  
        JSONParser jsonParser=new JSONParser();  
        try {  
            JSONObject packet=(JSONObject)jsonParser.parse(msg);  
            String cmd = (String)packet.get("cmd");  
            if(cmd.equals("login")) { //로그인 정보가 전송되어 오면  
                chatServer.area.append("클라이언트가 로그인 정보를 보냈습니다\n");  
                member = new Member(); //empty 상태의 VO 생성  
                //member VO에 정보 입력  
                JSONObject obj=(JSONObject)packet.get("member");  
                member.setMember_id(Integer.parseInt((String)obj.get("member_id")));  
                member.setUser_name((String)obj.get("user_name"));  
                member.setUser_id((String)obj.get("user_id"));  
                member.setUser_pass((String)obj.get("user_pass"));  
                member.setRegdate((String)obj.get("regdate"));  
                member.setImg((String)obj.get("img"));  
                StringBuilder sb = new StringBuilder();  
  
                sb.append("{");  
                sb.append("\"cmd\" : \"\n");  
                sb.append("\"memberList\" : [");  
                for(int i=0; i<chatServer.clientList.size(); i++) { //백터의 크기만큼 돌리려구요  
                    Member member = chatServer.clientList.get(i).member;  
                    sb.append("{");  
                    sb.append("\"member_id\" : \""+member.getMember_id()+"\"");  
                    sb.append("\"user_name\" : \""+member.getUser_name()+"\"");  
                    sb.append("\"user_id\" : \""+member.getUser_id()+"\"");  
                    sb.append("\"user_pass\" : \""+member.getUser_pass()+"\"");  
                    sb.append("\"regdate\" : \""+member.getRegdate()+"\"");  
                    sb.append("\"img\" : \""+member.getImg()+"\"");  
                    if(i<chatServer.clientList.size()-1) {  
                        sb.append(",");  
                    } else {  
                        sb.append("}");  
                    }  
                }  
                sb.append("]");  
                sb.append("}");  
  
                for(int i=0; i<chatServer.clientList.size(); i++) {  
                    ServerMsgThread smt=chatServer.clientList.get(i);  
                    smt.send(sb.toString()); //broadcast  
                }  
            }  
        }  
    }  
}
```

받은String을 제이슨으로 파싱 cmd분석 후 회원정보VO에 담고 스트링으로
다시
만들어 현재 클라이언트들에게 전송(브로드캐스팅)

전송 받은 String을 제이슨 으로 파싱, cmd분석 후 회원정보 VO에 담고
메인 클라이언트에 정보 전달+실시간 접속 인원 나타내기

채팅

클라이언트에서 실시간 접속 나타내기



- ◆ 채팅클라이언트에서 전송 시 채팅클라이언트가 아닌 메인을 통해서 서버와 통신한다 채팅은 더미일 뿐 기능은 메인 쪽에서 한다

ChatClient

```
//채팅클라이언트는 더미일뿐 서버와 통신하는건 메인프레임인 자신 //메시지 입력 엔터처리
t_input.addKeyListener(new KeyAdapter() {
    public void keyReleased(KeyEvent e) {
        if(e.getKeyCode() == KeyEvent.VK_ENTER) { //엔터치면..
            String msg = t_input.getText();
            appMain.sendMsg(msg);
            t_input.setText("");
        }
    }
});
```

- ◆ 채팅을 매개변수로 담은 후 메인에 있는 메서드로 전달

Main

```
// 메시지전송메서드
public void sendMsg(String msg) {
    this.msg=msg;
    StringBuilder sb = new StringBuilder();
    sb.append("{");
    sb.append("\"cmd\": \"chat\", ");
    sb.append("\"user_name\": \"\"+member.getUser_name()+"\"");
    sb.append("\"message\": \"\"+msg+"\"");
    sb.append("}");

    msgThread.send(sb.toString());
}
```

- ◆ 매개변수로 받은 내용을 cmd:chat 제이슨형식의 String으로 담아 클라이언트스레드로 전달

ClientServerThread

```
public void send(String msg) {
    try {
        buffw.write(msg+"\n");
        buffw.flush();
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

- ◆ 메인에게서 받은 String을 서버스레드로 전송

오픈 채팅방

ClientServerThread

```
public void listen() { // 끊임없이 듣기 위한 리스너 객체
    String msg=null;
    try {
        msg= buffr.readLine();
        JSONParser jsonParser=new JSONParser();

    } else if(cmd.equals("chat")) { // 채팅정보가 전송되어 오면
        String message = (String)packet.get("message");
        String user_name = (String)packet.get("user_name");
        appMain.chatClient.area.append(user_name+"님의 말 : "+message+"\n");
    }
}
```

- ◆ 받은String을 제이슨으로 파싱, cmd분석 후 채팅 클라이언트에 나타내기

ServerThread

```
public void listen() {
    String msg=null;
    try {
        msg= buffr.readLine(); // 클라이언트가 보낸 msg 이 포함된 스트링빌더(=sb(String))를 받아와 파싱을 한다
        //JSON 파싱 시작!!!
        JSONParser jsonParser=new JSONParser();
        try {
            JSONObject packet=(JSONObject)jsonParser.parse(msg);
            String cmd = (String)packet.get("cmd");

        } else if(cmd.equals("chat")) { // 대화의 메시지가 전송되어 오면
            String message = (String)packet.get("message");
            String user_name = (String)packet.get("user_name");
            StringBuilder sb = new StringBuilder();
            sb.append("{");
            sb.append("\"cmd\": \"chat\", ");
            sb.append("\"user_name\": \""+user_name+"\"");
            sb.append("\"message\": \""+message+"\"");
            sb.append("}");

            //대화할때 broadcasting !!!
            for(int i=0; i<chatServer.clientList.size(); i++) {
                ServerMsgThread msgThread=chatServer.clientList.get(i);
                msgThread.send(sb.toString());
            }
            chatServer.area.append(member.getUser_name()+"의 말 : "+message+"\n"); //area에 로그 남기기
        }
    }
}
```

- ◆ 받은String을 제이슨으로 파싱, cmd분석 후 String으로 변환 해 개인 정보와 함께 브로드 캐스팅 으로 현재 로그인한 인원에게 전송