



PAYDAY 앱보고서

<개발 완료 보고서>

팀명	MI A COL
팀원	신준호(팀장), 권택상, 이동규
최종작성일	2020년 08월 18일

목 차

1. 제안

- 1.1 팀 구성 및 역할 선정
- 1.2 회의 시간 및 방식 선정
- 1.3 문제 인식 및 주제 선정

2. 구상 & 분석

- 2.1 프로젝트 목표 인식
- 2.2 콘텐츠 개요 작성

3. 설계

- 3.1 프로젝트 목표 설정
- 3.2 프로그램 개발 방향 설정
- 3.3 개발 툴 학습 – Android studio, JAVA

4. 구현 및 메뉴얼

- 4.1 디자인
- 4.2 개발 환경 디자인 구현
- 4.3 구현내용

5. 테스트

- 5.1 단위 테스트 및 통합 테스트

6. 완료

- 6.1 구글 플레이스토어 등록
- 6.2 프로젝트 완료 소감

부록

- 부록-1 자바코드
- 부록-2 xml코드

1.1 팀 구성 및 역할 선정

프론트엔드(디자인) : 신준호

백엔드(데이터 베이스, 연산처리코드) 권택상, 신준호

앱 영상, 앱 아이콘 : 이동규

도움 : 이원효(버튼 이미지)

1.2 회의 시간 및 방식 설정

회의 시간 : 매주 **60분** 이상 회의가 길어진다면 가능한 만큼 최대한 길게

회의 수단 : 팀즈 및 필요에 따라 대면 회의

1차 회의	2020.04.16	주제선정, 요구사항 분석
2차 회의	2020.04.24	앱아이디어 why, how, what 작성 및 아이디어 구체화
3차 회의	2020.05.03	앱 구현시 사용 툴 및 언어 설정
4차 회의	2020.05.09	앱 구현 역할 분담, 디자인 초안 다지기
5차 회의	2020.05.14	사용 데이터 베이스 결정
6차 회의	2020.06.24	팀장 1명 중간 탈퇴로 인해 팀장 교체 앱 디자인 구체화 및 앱 실행 과정 재구체화, 재 역할선정, 새로운 인원모집, 앱기능 재구체화 및 아이디어 재구체화 완료
7차 회의	2020.06.30	앱 제작시작 및 앱 툴 공부 시작
8차 회의	2020.07.01	구현 가능한 디자인 범위 설정 및 디자인 회의
9차 회의	2020.07.15	공동작업 및 진행 과정 보고
10차 회의	2020.07.22	테스트, 공동작업 및 진행 과정 보고
11차 회의	2020.08.05	앱 튜토리얼 영상제작 회의, 앱 테스트, 공동작업 및 진행 과정 보고
12차 회의	2020.08.12	앱 튜토리얼 영상제작 및 디버그
13차 회의	2020.08.15	앱 구글플레이스토어 등록

1.3 문제 인식 및 주제선정

1. WHY

- 현존하는 어플이나 프로그램에서는 주휴수당, 야간수당 등등 여러 항목을 복합적으로 계산 할 수 있는 프로그램이 없다.
- 문제가 발생할 경우 자신이 근무를 했다는 증거물이 필요하다.

2. WHAT

임금 계산기 - 원하는 기간을 설정하여 그 기간의 법정근로시간에 알맞은 임금을 계산해준다.

※ 통상임금, 주휴수당, 연장수당, 야간근로수당, 주휴수당

3. HOW

android studio(JAVA) 사용

- 임금 계산

캘린더에서 사진기록, 시간등록을 할 수 있다.(사진에는 찍힐 때마다 시간기록이 데이터 베이스에 남고 사진이름도 시간기록으로 설정된다.)

- 기간을 설정하면 그 기간에 해당하는 임금을 계산을 할 수 있다.

- 데이터베이스에 보관된 선택한 아르바이트에 대한 모든 기록을 삭제하거나 기록을 잊을 걱정 없이 엑셀 파일로 만들 수 있다.

2. 구상 & 분석

2.1 프로젝트 목표 인식

아르바이트를 하는 모든 근로자가 좀 더 쉽게 월급을 계산할 수 있도록 합니다. 아르바이트 중 부당한 일을 당했을 때 증거물 파일을 이용하여 이를 방지할 수 있습니다. 이러한 기능들이 근로자가 더 쉽게 권리를 요구할 수 있도록 합니다.

2.2 콘텐츠 개요 작성

메인화면	
알바계산기	알바 정보(기간, 시급, 이름, 특이사항)를 입력할 시 원하는 기간만큼의 금액을 계산하여 보여준다.
달력	당일의 출·퇴근 기록과 메모를 저장할 수 있다.
증거물 파일 생성 버튼	저장된 모든 내용을 엑셀 파일로 저장할 수 있다.
파일 경로 버튼	저장된 엑셀 파일과 출·퇴근 인증사진의 파일 경로를 알 수 있다.
알바 선택해 버튼	
추가(+)버튼	버튼을 누를 시 알바 이름, 시급, 날짜, 근무시간, 5인 이상 사업장, 무급 휴식시간을 설정할 수 있다.
수정 설정	수정을 활성화하여 알바 이름을 누를 시 알바 정보를 수정할 수 있다.
선택 설정	선택을 활성화하여 알바 이름을 누를 시 이름에 맞는 정보가 달력에 표시된다.
삭제 설정	삭제를 활성화하여 알바 이름을 누를 시 알바 정보가 삭제됩니다.

3. 설계

3.1 프로젝트 목표 설정

저희 팀은 이번에 사람들이 필요한 것이 무엇일까 또는 우리가 필요한 것은 무엇일까에 중점을 두고 이 프로젝트를 계획하였습니다. 그 결과 아르바이트 계산기라는 아이디어를 생각하였고 저희 앱이 다른 비슷한 앱들과의 차별점을 두기 위하여 저희가 원하는 목표인 사람들의 급여의 권리를 지키기 위한 앱을 구현하기로 결정하였습니다.

프로젝트의 개발은 각자 역할을 분담하고 이후 내용들을 전체적으로 병합하여 최종적으로는 전체적인 코드의 오류 발생을 처리하고 개발된 프로그램을 문서화하는 작업을 거치는 것을 목표로 하였습니다

3.2 프로그램 개발 방향 설정

로그인도 필요없고 간편한 아르바이트 급여 계산 및 증거물 기록 및 증거물 생성 앱 제작
앱 제작시 디자인도 깔끔하고 사용자가 편하게 받아들일 수 있게 제작

필요 이유	아르바이트를 할 시 급여를 적게 받거나 부당한 대우를 받을 수가 있다. 그러한 것들을 앱에 기록하여 아르바이트를 하는 모든 사용자의 권리를 지킬 수 있도록 도와주는 앱
개발 방향	기간은 일주일, 월 단위가 아닌 자유롭게 선택할 수 있게 한다 출근과 퇴근시 출근과 퇴근을 했다는 인증사진을 남길 수 있도록 한다. 여러개의 아르바이트를 하더라도 아르바이트 기록에 문제없게 한다. 증거물을 바로 생성할 수 있도록 한다. 쉽고 편한 사용자 인터페이스를 사용한다. 야근수당, 연장수당, 주휴수당, 주말수당에 관해서는 따로 선택없이 자동으로 계산해준다.

3.3 개발 툴 학습 – 안드로이드 스튜디오, JAVA

안드로이드 스튜디오와 자바의 경우에 한번도 해본 경험이 없어 하루만에 끝낼 수 있는 짧은 기초책으로 공부한 후 앱을 제작하며 모르는 부분은 **stack overflow** 및 여러 카페들을 참조해가며 앱을 제작했습니다.

4. 구현 및 메뉴얼

개발 환경

- 안드로이드 스튜디오

< 디자인 >

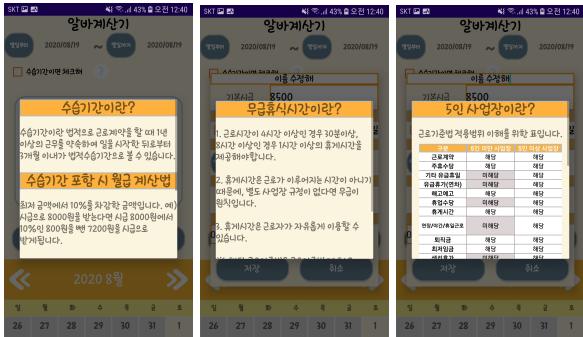


< 구현 내용 >

 main	<p>몇시부터 몇시까지 버튼</p> <p>원하는 근무 기간을 설정하여 그 기간동안 벌어들인 금액을 계산하는데 사용됩니다.</p>
	<p>수습기간 체크박스</p> <p>활성화 시 수습기간에 관한 근로법에 따른 월급의 10%를 차감한 금액을 받도록 계산해줍니다.</p>
	<p>알바선택해 버튼</p> <p>원하는 아르바이트의 정보를 추가, 수정, 삭제, 선택할 수 있는 버튼입니다.</p>
	<p>캘린더</p> <p>알바선택해 버튼에서 선택한 알바에 관한 일정만 표시되어 그 날들에 대해 출·퇴근 인증을 할 수 있습니다.</p>
	<p>증거물 파일 생성 버튼</p> <p>출·퇴근 인증 이미지 파일, 근무 일시, 메모 등의 아르바이트 기록을 엑셀 파일로 저장해줍니다.</p>
	<p>파일 경로 버튼</p> <p>저장된 이미지와 엑셀 파일의 경로를 보여줍니다.</p>

 알바 선택해 버튼 누를 시	<p>추가(+) 버튼</p> <p>알바 목록에 알바 버튼을 생성합니다.</p>
	<p>수정</p> <p>활성화 후 알바 버튼 클릭 시 알바 정보를 수정할 수 있습니다.</p>
	<p>선택</p> <p>활성화 후 알바 버튼 클릭 시 창이 닫히고 캘린더에 선택한 알바의 일정이 표시됩니다.</p>
	<p>삭제</p> <p>활성화 후 알바 버튼 클릭 시 삭제됩니다.</p>
	<p>추가로 생성된 알바 버튼 클릭 시 나오는 창</p> <p>이름 수정, 시급, 요일, 시간, 5인 사업장, 휴식 시간들에 관한 정보를 수정할 수 있습니다.</p>

 캘린더 누를 시	<p>몇일부터 몇일까지</p> <p>야간 수당과 추가 근무수당 등을 계산하기 위해 당일만 일하고 끝나는지 다음 날 까지 일하는지 사용자가 설정하여 저장할 수 있습니다.</p>
	<p>몇시부터 몇시까지</p> <p>근무시간을 사용자가 설정하여 저장할 수 있습니다.</p>
	<p>메모</p> <p>사용자가 메모할 내용을 기입할 수 있습니다.</p>
	<p>출·퇴근 인증 사진</p> <p>출근 기록과 퇴근 기록을 이미지로 표시하여 저장이 가능합니다.</p>

 <p style="text-align: center;">도움말</p>	<p>수습기간 도움말</p>	<p>수습기간에 대한 설명과 수습기간을 계산하는 방법이 안내되어 있습니다.</p>
	<p>무급휴식시간 도움말</p>	<p>근로기준법에 의한 무급 휴식 시간 기준이 안내되어 있습니다.</p>
	<p>5인 사업장 도움말</p>	<p>5인 이상 사업장 또는 5인 미만 사업장에 대한 내용과 계산 방법이 표와 설명으로 안내되어 있습니다.</p>

5. 테스트

5.1 단위 테스트 및 통합 테스트

9차 회의	2020.07.15	디자인 및 프론트 엔드 부분 테스트 및 디버깅
10차 회의	2020.07.22	데이터베이스 테스트 및 디버깅
11차 회의	2020.08.05	프론트 엔드와 백엔드가 연결되는 부분 테스트 및 디버깅
12차 회의	2020.08.12	전체 테스트 및 급여연산오류, 사진위치, 데이터베이스 오류 디버깅
13차 회의	2020.08.15	앱 구글플레이스토어 등록전 전체 테스트 10회 및 급여 연산오류 및 데이터베이스 오류 디버깅

6. 완료

6.1 구글 플레이스토어 등록

상품 세부정보

한국어 - KO-KR

번역 관리 ▾

제목 *

한국어 – ko-KR

Payday - 알바 계산기 & 알바 기록 앱

25/50

간단한 설명 *

한국어 – ko-KR

일하는 모든 사람들을 위한 알바계산기 & 알바기록 앱 서비스

34/80

자세한 설명 *

한국어 – ko-KR

※ 주휴수당, 연장수당, 야간수당, 수습기간 내 수당, 주말수당을 자동으로 적용하여 계산합니다.※

4. 증거물 파일 생성

- 근무 시 부당한 일을 겪었을 때 증거물로 제시할 수 있는 모든 근무 기록을 저장한 엑셀파일을 생성합니다.
- 엑셀파일 저장은 메인화면 ->'증거물 파일 생성' -> '예' 버튼을 누르시면 됩니다.
- ※ 파일저장 위치는 어플리케이션 메인화면 ->'파일경로' 버튼을 누르시면 됩니다.

10 이사하나

760/4000

Google의 [메타데이터 정책](#)을 확인하여 앱 메타데이터와 관련된 몇 가지 자주 발생하는 위반을 방지하시기 바랍니다. 또한 기타 모든 [프로그램 정책](#)을 검토한 다음 앱을 제출해 주세요.

앱 또는 스토어 등록정보를 Google Play 앱 검토팀에 사전에 알릴 수 있는 경우 [Google에 문의한](#) 다음 게시해 주세요.

그래픽 저작물

앱 아이콘, 스크린샷, 동영상을 관리하여 Google Play에서 앱을 홍보하세요. 자세히 알아보기

새로운 그래픽을 업로드하기 전에 [명의 도용 및 지적 재산권 정책](#)을 검토하세요. 현지화된 그래픽 없이 스토어 등록정보에 번역을 추가하면 기본 등록정보의 그래픽이 사용됩니다.

고해상도 아이콘 *

기본 언어 – 한국어 – ko-

KR

512x512

32비트 PNG

Google Play 아이콘 사양

Google Play에서 아이콘에 등근 모서리 및 그림자를 등적으로 적용합니다. 투명 배경은 더 이상 허용되지 않습니다. 자세히 알아보기



Google Play에 표시되는 아이콘

스크린샷 *

기본 언어 – 한국어 – ko-KR

JPEG 또는 24비트 PNG(알파 없음). 측면 최소 길이: 320픽셀. 측면 최대 길이: 3840픽셀.

전체적으로 최소 2개의 스크린샷이 필요합니다. 유형별 스크린샷은 최대 8개입니다. 다시 정렬하거나 유형 간에 이동하려면 드래그하세요.

스크린샷 4/8개 파일 찾아보기

그래픽 이미지 *

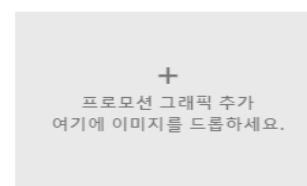
기본 언어 – 한국어 – ko-KR
너비x높이 1024x500
JPG 또는 24비트 PNG(알파 없음)

프로모션 그래픽

기본 언어 – 한국어 – ko-KR
너비x높이 180x120
JPG 또는 24비트 PNG(알파 없음)

TV 배너

기본 언어 – 한국어 – ko-KR
너비 1280 x 높이 720
JPG 또는 24비트 PNG(알파 없음)



Daydream 360° 입체 영상 이미지

기본 언어 – 한국어 – ko-KR
너비 4096x높이 4096
JPG 또는 24비트 PNG(알파 없음)

프로모션 동영상

기본 언어 – 한국어 – ko-KR
YouTube 등 영상
URL을 입력하세요.

<https://www.youtube.com/watch?v=ZY3yb9qY0Os&feature=youtu.be>

출시: 1.0.0.3 수정
어제, 오후 10:34: 출시되었습니다.

출시 내역
어제, 오후 10:34: 출시되었습니다.

이번 버전의 새로운 기능
기본 언어 – 한국어 – ko-KR
출시직후
안드로이드 버전호환 문제를 수정했습니다.

※ 1개 언어로 번역됨 출시 노트 수정

Android App Bundle 및 APK
 이 버전은 Google Play 64비트 요구사항을 충족합니다. 자세히 알아보기

[모두 펼치기](#)

유형	버전 코드	API 수준	업로드됨	앱 다운로드 크기	사용 중인 기기에 설치된 횟수
App Bundle 1개가 추가됨	Android App Bundle 3	14+	어제, 오후 10:30	9.05~9.09MB	데이터 없음
App Bundle 1개가 비활성화됨					

출시 내역

프로젝트 완료 소감

인원	소감내용
신준호	<p>처음으로 저희의 첫 앱을 만들어서 너무 기쁩니다. 프로젝트 과정은 팀장이 중간에 나가고 제가 팀장이 되어 순탄치는 않았으나 모두 열심히 해주어서 정말 고마웠습니다. 안드로이드 앱을 만드는 것이 처음이라 처음부터 하나하나 직접 찾아서 익히고 그러한 처음 했던 경험이 저의 프로그래밍 실력 향상과 실제 구현에 관련한 경험이 늘어서 다음에 다른 앱을 구현할 때는 조금 더 복잡한 앱도 만들 수 있다는 그 다음 계획에 대한 의지가 생겼습니다. 앱을 만드는 것이 어려웠지만 즐거웠고 결국 처음으로 앱을 등록하게 되어 정말 기분이 좋습니다. 이번년도 이 멤버들로 조금 더 많은 프로그램들을 만들 계획인데 자신감이 더 불어서 이번 프로젝트는 성공적이었다고 생각합니다.</p>
권택상	<p>프로젝트에 중간부터 참여하게 되었을 때 우왕좌왕하다 팀원들에게 큰 도움이 되지 못할까봐 걱정했지만 팀원인 준호가 자료 도움과 질의응답을 잘 해 줘서 새로운 지식들을 습득하는데 시간 절약을 할 수 있었습니다. 그리고 직접 만나서 한 프로젝트 회의에서는 팀원인 동규형의 경험과 지식으로 프로젝트에 대해 저와 다른 학과에서 본 새로운 견해와 조언을 얻거나 문제해결을 하는 방법을 배울 수 있었습니다.</p> <p>프로젝트 중 앱에서 입력한 정보를 필요한 정보들만 정리해서 데이터베이스에 입력하는 법에 대해서 배웠고 정리한 데이터베이스 자료를 파일 형식으로 바꾸는 법도 배울 수 있었습니다. 이와 달리 프론트 엔드라는 새로운 분야를 직접적으로 접하게 되면서 이에 흥미를 가지는 계기가 되었습니다.</p> <p>이 프로젝트를 하면서 프로그래밍 지식은 끝이 없음을 현실로써 알게 되었고 앞으로 새로운 프로그래밍 지식에 더 정진할 계획입니다.</p>
이동규	<p>우선 팀원들과 함께한 APP이 제작되어 출시되었다는 사실에 감격스럽습니다. 개인적인 사정으로 개발을 같이 진행하지 못 한 부분이 매우 아쉬웠으나 아이콘 및 사용설명서 영상을 만들고 팀원들에게 도움을 줄 수 있어서 한편으로는 다행이라고 생각합니다. 어플리케이션이 아직 초기 단계지만 안드로이드 학습을 통해 같이 진행하도록 노력하겠습니다.</p>

부록-1(java 코드)

AlertDialog_addxls.java

```
package com.alba.lionproject1;

import android.database.Cursor;
import android.os.Bundle;

import android.widget.Button;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.CellStyle;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.*;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;

/* 증거율 파일 생성 버튼 이벤트 코드입니다.
사용안내와 예를 누르면 엑셀 파일이 생성되고 아니요를 누르면 창을 닫습니다.*/
public class AlertDialog_addxls extends AppCompatActivity {
    Button AddXlsYes;
    Button AddXlsNo;
    Cursor cursor;
    String[][] registerArray = new String[100][9];
    Button albabutton = ((MainActivity) MainActivity.context_main).findViewById(R.id.btn_ChOOSEParttime);

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.alertdialog_addxls);

        TextView xlsExplain = findViewById(R.id.txt_XlsExplain);
        String text = "엑셀 파일 경로는 메인화면의 파일경로 버튼을 눌러서 확인할 수 있어\n\n" +
                "※ 파일은 부당한 일이 생겼을 때 필요할 거야";
        xlsExplain.setText(text);
```

```

// DB 데이터들을 배열에 저장하고 그 데이터들로 엑셀 파일을 만드는 버튼
AddXlsYes = findViewById(R.id.btn_XlsYes);
AddXlsYes.setOnClickListener(view -> {
    // DB 데이터를 배열에 저장하는 코드
    cursor = getregisterCursor();
    // row
    for (int i = 0; i < cursor.getCount(); i++) {
        cursor.moveToPosition(i);
        // column
        for (int j = 0; j < 9; j++) {
            registerArray[i][j] = cursor.getString(j);
        }
    }
}

// 배열에 저장된 내용을 Run창에 보여주는 코드(개발자 확인용)
// row
for (int i = 0; i < cursor.getCount(); i++) {
    cursor.moveToPosition(i);
    // column
    System.out.println(registerArray[i][0] + " " + registerArray[i][1] + " " +
        registerArray[i][2] + "\t" + registerArray[i][3] + "\t" +
        registerArray[i][4] + "\t" + registerArray[i][5] + "\t" +
        registerArray[i][6] + "\t" + registerArray[i][7] + "\t" +
        registerArray[i][8]);
}
}

// 엑셀 파일에 내용을 입력해서 엑셀 파일 만들기
Workbook wb = new HSSFWorkbook();
Cell cell;

Sheet sheet;
sheet = wb.createSheet("Name of sheet");

CellStyle cs = wb.createCellStyle();
cs.setWrapText(true);

Row row = sheet.createRow(0);

cell = row.createCell(0);
cell.setCellStyle(cs);
cs.setAlignment(HorizontalAlignment.CENTER);
cell.setCellValue("DATE_FROM");

cell = row.createCell(1);
cell.setCellStyle(cs);
cs.setAlignment(HorizontalAlignment.CENTER);
cell.setCellValue("TITLE");

cell = row.createCell(2);
cell.setCellStyle(cs);
cs.setAlignment(HorizontalAlignment.CENTER);
cell.setCellValue("FROM");

cell = row.createCell(3);
cell.setCellStyle(cs);
cs.setAlignment(HorizontalAlignment.CENTER);
cell.setCellValue("TO");

```

```

cell = row.createCell(4);
cell.setCellStyle(cs);
cs.setAlignment(HorizontalAlignment.CENTER);
cell.setCellValue("MEMO");

cell = row.createCell(5);
cell.setCellStyle(cs);
cs.setAlignment(HorizontalAlignment.CENTER);
cell.setCellValue("IMG_AT");

cell = row.createCell(6);
cell.setCellStyle(cs);
cs.setAlignment(HorizontalAlignment.CENTER);
cell.setCellValue("IMG_AFTER");

cell = row.createCell(7);
cell.setCellStyle(cs);
cs.setAlignment(HorizontalAlignment.CENTER);
cell.setCellValue("DATE_TO");

for (int i = 0; i < cursor.getCount(); i++) { // 데이터 엑셀에 입력
    row = sheet.createRow(i + 1);
    for (int j = 1; j < 9; j++) {
        cell = row.createCell(j - 1);
        if (String.valueOf(registerArray[i][j]).equals(albabutton.getText().toString())) {
            cell.setCellValue(String.valueOf(registerArray[i][j]));
            System.out.println("registerArray = " + registerArray[i][j]);
        }
        cell.setCellStyle(cs);
        cs.setAlignment(HorizontalAlignment.CENTER);
        cs.setVerticalAlignment(VerticalAlignment.CENTER);
        sheet.setColumnWidth(j - 1, (5000));
    }
}

File file = new File(getExternalFilesDir(null), "plik.xls");
FileOutputStream outputStream = null;
try {
    outputStream = new FileOutputStream(file);
    wb.write(outputStream);
    Toast.makeText(getApplicationContext(), "ok", Toast.LENGTH_LONG).show();
} catch (IOException e) {
    e.printStackTrace();
    Toast.makeText(getApplicationContext(), "so ok", Toast.LENGTH_LONG).show();
    try {
        assert outputStream != null;
        outputStream.close();
    } catch (IOException ex) {
        ex.printStackTrace();
    }
}
});

// 아니요 버튼(나가기)
AddXlsNo = findViewById(R.id.btn_XlsNo);
AddXlsNo.setOnClickListener(view -> finish());
}

```

```

// DB 데이터 읽어오기
private Cursor getregisterCursor() {
    registerWorkDBHelper db = registerWorkDBHelper.getInstance(this);
    return db.getReadableDatabase().query(
        registerWorkContract.registerWorkEntry.TABLE_NAME, null, null, null, null, null,
        registerWorkContract.registerWorkEntry._ID);
}
}

```

AlertDialog_filepath.java

```

package com.alba.lionproject1;

import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class AlertDialog_filepath extends AppCompatActivity {
    Button Close;
    String imgPath = "1. file:/storage/emulated/0/Android/data/com.alba.lionproject1/files/Pictures\n\n"
+
        "2. 내 파일 > 내장 메모리 > Android > data > com.alba.lionproject1 > files > Pictures";
    String xlsPath = "1. file:/storage/emulated/0/Android/data/com.alba.lionproject1/files\n\n" +
        "2. 내 파일 > 내장 메모리 > Android > data > com.alba.lionproject1 > files";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.alertdialog_filepath);

        TextView imgfilepath = findViewById(R.id.txt_imgfilepath);
        imgfilepath.setText(imgPath);
        TextView xlsfilepath = findViewById(R.id.xlsfilepath);
        xlsfilepath.setText(xlsPath);

        Close = findViewById(R.id.btn_Close);
        Close.setOnClickListener(view -> finish());
    }
}

```

customcalendarview.java

```

package com.alba.lionproject1;

import android.content.Context;
import android.content.Intent;

```

```

import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.util.AttributeSet;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.GridView;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import android.widget.TextView;
import android.widget.Toast;

import androidx.annotation.Nullable;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import java.util.Locale;

public class customcalendarview extends LinearLayout {
    ImageButton NextButton, PreviousButton;
    TextView CurrentDate;
    GridView gridView;

    private static final int MAX_CALENDAR_DAYS = 42;
    public static Calendar calendar = Calendar.getInstance(Locale.KOREA);
    public static Context context;
    public static SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy MMMM", Locale.KOREA);
    public static SimpleDateFormat monthFormat = new SimpleDateFormat("MMMM", Locale.KOREA);
    public static SimpleDateFormat yearFormat = new SimpleDateFormat("yyyy", Locale.KOREA);
    public static SimpleDateFormat eventDateFormat = new SimpleDateFormat("yyyy-MM-dd",
    Locale.KOREA);

    MyGridAdapter myGridAdapter;
    List<Date> dates = new ArrayList<>();
    List<Events> eventsList = new ArrayList<>();

    DBOpenHelper dbOpenHelper;

    public customcalendarview(Context context) {
        super(context);
    }

    public customcalendarview(final Context context, @Nullable AttributeSet attrs) {
        super(context, attrs);
        customcalendarview.context = context;
        InitializeLayout();
        SetUpCalendar();

        String currentMonthAndYear = (String) CurrentDate.getText();
        PreferenceManager.setString(context, "time", currentMonthAndYear);

        PreviousButton.setOnClickListener(v -> {
            calendar.add(Calendar.MONTH, -1);
            SetUpCalendar();
        });
    }
}

```

```

        NextButton.setOnClickListener(view -> {
            calendar.add(Calendar.MONTH, 1);
            SetUpCalendar();
        });

        // 그리드 즉 날짜칸 클릭했을 때 발생하는 이벤트
        gridView.setOnItemClickListener((parent, view, position, id) -> {
            Button btn_parttimeselect = ((MainActivity)
                MainActivity.context_main).findViewById(R.id.btn_ChooseParttime);
            if (!btn_parttimeselect.getText().toString().equals("알바 선택해")) {
                final String date = eventDateFormat.format(dates.get(position));
                final String month = monthFormat.format(dates.get(position));
                final String year = yearFormat.format(dates.get(position));
                //SaveEvent("event", "time", date, month, year);
                //SetUpCalendar();
                Intent intent = new Intent(getApplicationContext(), registerWork.class);
                intent.putExtra("date", date);
                intent.putExtra("month", month);
                intent.putExtra("year", year);
                context.startActivity(intent);
            } else {
                Toast.makeText(getApplicationContext(), "알바 기본정보를 설정하고 그 알바를 선택해야 들어갈 수 있어",
                    Toast.LENGTH_SHORT).show();
            }
        });
    }

    public customcalendarview(Context context, @Nullable AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);

    }

    public void InitializeLayout() {
        LayoutInflater inflater = (LayoutInflater)
        context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View view = inflater.inflate(R.layout.calendar_layour, this);
        NextButton = view.findViewById(R.id.nextBtn);
        PreviousButton = view.findViewById(R.id.previousBtn);
        CurrentDate = view.findViewById(R.id.current_Date);
        gridView = view.findViewById(R.id.gridView);
    }

    public void SetUpCalendar() {
        String currentDate = dateFormat.format(calendar.getTime());
        CurrentDate.setText(currentDate);
        dates.clear();
        Calendar monthCalendar = (Calendar) calendar.clone();
        monthCalendar.set(Calendar.DAY_OF_MONTH, 1);
        int FirstDayOfMonth = monthCalendar.get(Calendar.DAY_OF_WEEK) - 1;
        monthCalendar.add(Calendar.DAY_OF_MONTH, -FirstDayOfMonth);
        CollectEventsPerMonth(monthFormat.format(calendar.getTime()),
            yearFormat.format(calendar.getTime()));

        while (dates.size() < MAX_CALENDAR_DAYS) {
            dates.add(monthCalendar.getTime());
        }
    }
}

```

```

monthCalendar.add(Calendar.DAY_OF_MONTH, 1);

}

myGridAdapter = new MyGridAdapter(context, dates, calendar, eventsList);
gridView.setAdapter(myGridAdapter);

}

private void CollectEventsPerMonth(String Month, String year) {
    eventsList.clear();
    dbOpenHelper = new DBOpenHelper(context);
    SQLiteDatabase database = dbOpenHelper.getReadableDatabase();
    Cursor cursor = dbOpenHelper.ReadEventsperMonth(Month, year, database);
    while (cursor.moveToNext()) {
        String event = cursor.getString(cursor.getColumnIndex(DBStructure.EVENT));
        String time = cursor.getString(cursor.getColumnIndex(DBStructure.TIME));
        String date = cursor.getString(cursor.getColumnIndex(DBStructure.DATE));
        String month = cursor.getString(cursor.getColumnIndex(DBStructure.MONTH));
        String Year = cursor.getString(cursor.getColumnIndex(DBStructure.YEAR));
        Events events = new Events(event, time, date, month, Year);
        eventsList.add(events);
    }
    cursor.close();
    dbOpenHelper.close();
}

}

```

DataBases.java

```

package com.alba.lionproject1;

import android.provider.BaseColumns;

public class DataBases {

    public static final class CreatDB implements BaseColumns {
        public static final String _TABLENAME0 = "usertable";
        public static final int DB_VERSION = 1;
        public static final String PARTTIMENAME = "parttimename";
        public static final String HOURLYWAGE = "wage";
        public static final String DAYOFWEEK = "dayofweek";
        public static final String WORKINGSTART = "workingstart";
        public static final String WORKINGFINISH = "workingfinish";
        public static final String NUMBER_OF_PEOPLE = "numberofpeople";
        public static final String MEAL_TIME = "mealtime";
        public static final String _DROP0 = "DROP TABLE IF EXISTS " + _TABLENAME0;
        public static final String _CREATE0 = "CREATE TABLE IF NOT EXISTS " + _TABLENAME0 + "("
            + _ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "
            + PARTTIMENAME + " TEXT NOT NULL , "
            + HOURLYWAGE + " TEXT NOT NULL , "
            + DAYOFWEEK + " TEXT NOT NULL , "
            + WORKINGSTART + " TEXT NOT NULL , "
            + WORKINGFINISH + " TEXT NOT NULL , "
            + NUMBER_OF_PEOPLE + " TEXT NOT NULL , "
            + MEAL_TIME + " TEXT NOT NULL );";
    }
}

```

```
}
```

DBOpenHelper.java

```
package com.alba.lionproject1;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class DBOpenHelper extends SQLiteOpenHelper {

    private static final String CREATE_EVENTS_TABLE = "create table " +
DBStructure.EVENT_TABLE_NAME + "(ID INTEGER PRIMARY KEY AUTOINCREMENT, "
        + DBStructure.EVENT + " TEXT, " + DBStructure.TIME + " TEXT, " + DBStructure.DATE + " TEXT,
" + DBStructure.MONTH + " TEXT, "
        + DBStructure.YEAR + " TEXT)";
    private static final String DROP_EVENTS_TABLE = "DROP TABLE IF EXISTS " +
DBStructure.EVENT_TABLE_NAME;

    public DBOpenHelper(@Nullable Context context) {
        super(context, DBStructure.DB_NAME, null, DBStructure.DB_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(CREATE_EVENTS_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL(DROP_EVENTS_TABLE);
        onCreate(db);
    }

    public void SaveEvent(String event, String time, String date, String month, String year, SQLiteDatabase database) {
        ContentValues contentValues = new ContentValues();
        contentValues.put(DBStructure.EVENT, event);
        contentValues.put(DBStructure.TIME, time);
        contentValues.put(DBStructure.DATE, date);
        contentValues.put(DBStructure.MONTH, month);
        contentValues.put(DBStructure.YEAR, year);

        database.insert(DBStructure.EVENT_TABLE_NAME, null, contentValues);
    }
}
```

```

public Cursor ReadEventsperMonth(String month, String year, SQLiteDatabase database) {
    String[] Projections = {DBStructure.EVENT, DBStructure.TIME, DBStructure.DATE,
    DBStructure.MONTH, DBStructure.YEAR};
    String Selection = DBStructure.MONTH + "=?" + " and " + DBStructure.YEAR + "=?";
    String[] SelectionArgs = {month, year};
    return database.query(DBStructure.EVENT_TABLE_NAME, Projections, Selection, SelectionArgs,
    null, null, null);
}
}

```

DBStructure.java

```

package com.alba.lionproject1;

public class DBStructure {
    public static final String DB_NAME = "EVENTS_DB";
    public static final int DB_VERSION = 1;
    public static final String EVENT_TABLE_NAME = "eventstable";
    public static final String EVENT = "event";
    public static final String TIME = "time";
    public static final String DATE = "date";
    public static final String MONTH = "month";
    public static final String YEAR = "year";
}

```

Events.java

```

package com.alba.lionproject1;

public class Events {
    String EVENT, TIME, DATE, MONTH, YEAR;

    public String getDATE() {
        return DATE;
    }

    public Events(String EVENT, String TIME, String DATE, String MONTH, String YEAR) {
        this.EVENT = EVENT;
        this.TIME = TIME;
        this.DATE = DATE;
        this.MONTH = MONTH;
        this.YEAR = YEAR;
    }
}

```

getDatePopUp.java

```

package com.alba.lionproject1;

import androidx.appcompat.app.AppCompatActivity;

```

```
import android.content.Context;
import android.content.Intent;

import android.database.Cursor;

import android.os.Bundle;
import android.view.View;

import android.view.WindowManager;

import android.widget.Button;
import android.widget.DatePicker;
import android.widget.EditText;

import android.widget.TextView;
import android.widget.Toast;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;

import java.util.Calendar;
import java.util.Date;
import java.util.GregorianCalendar;

import java.util.Locale;
import java.util.Objects;

public class getDatePopUp extends AppCompatActivity {
    ArrayList<String[]> hoursSubtractions = new ArrayList<>();
    ArrayList<String[]> workingHours = new ArrayList<>();
    double[] oneDayWage = new double[100000];
    String[][] totalArray = new String[100][8];
    String[][] registerArray = new String[100000][9];
    String NEXTDATE;
    String CURRENTDATE;
    Button btn_choose;
    public static String screenview = "";
    public String date;
    private int mYear = 0, mMonth = 0, mDay = 0; // 년 월 일 변수 생성후 초기화
    double totalWage = 0;
    double totalWorkingHours = 0;
    double[] extraMoney = new double[5];
    int firstWeek = 0;
    int secondWeek = 0;
    int thirdWeek = 0;
    int forthWeek = 0;
    int fifthWeek = 0;
    int count = 0;
    Myapplication myapplication;
    SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd", Locale.KOREA);
    SimpleDateFormat format = new SimpleDateFormat("yyyy/MM/dd", Locale.KOREA);
    SimpleDateFormat timeDateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm", Locale.KOREA);
    TextView txt_From;
    TextView txt_to;
    Date txtFrom;
    Date txtTo;
```

```

Context context;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_get_date_pop_up);
    getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
        WindowManager.LayoutParams.FLAG_FULLSCREEN);
    setContentView(R.layout.activity_get_date_pop_up);
    context = this;

    myapplication = new Myapplication();
    btn_choose = ((MainActivity) MainActivity.context_main).findViewById(R.id.btn_ChoosePartime);
    txt_From = ((MainActivity) MainActivity.context_main).findViewById(R.id.txt_from);
    txt_to = ((MainActivity) MainActivity.context_main).findViewById(R.id.txt_to);

    // 한국으로 국가설정
    Locale.setDefault(Locale.KOREA);
    this.setContentView(R.layout.activity_get_date_pop_up);

    //전에 눌렀던 버튼을 기준으로 화면을 구분해서 그에맞는 텍스트에 뿌려주기 위해 Intent를 사용하여
    전화면의 버튼의 정보와 화면정보 가져옴
    try {
        Intent intent = getIntent();
        screenview = Objects.requireNonNull(intent.getExtras()).getString("view");
        NEXTDATE = Objects.requireNonNull(intent.getExtras()).getString("nextdate");
        CURRENTDATE = Objects.requireNonNull(intent.getExtras()).getString("currentdate");
    } catch (Exception e) {
        e.printStackTrace();
    }

    try {
        switch (screenview) {
            case "mainfrom": {
                String[] date = txt_From.getText().toString().split("/");
                if (!date[0].equals("") || !date[1].equals("") || !date[2].equals("")) {
                    mYear = Integer.parseInt(date[0]);
                    mMonth = Integer.parseInt(date[1]) - 1;
                    mDay = Integer.parseInt(date[2]);
                }
                break;
            }
            case "mainto": {
                String[] date = txt_to.getText().toString().split("/");
                if (!date[0].equals("") || !date[1].equals("") || !date[2].equals("")) {
                    mYear = Integer.parseInt(date[0]);
                    mMonth = Integer.parseInt(date[1]) - 1;
                    mDay = Integer.parseInt(date[2]);
                }
                break;
            }
            case "rwdateto": {
                // 스피닝 캘린더(layout폴더에서 getdatepopup xml 파일 보시면 Datepicker라고 써있는데
                종류가 스피닝 캘린더입니다.)
                Calendar calendar = new GregorianCalendar();
                mYear = calendar.get(Calendar.YEAR); //스피닝 캘린더로 요번 년도값(year) 받기
                mMonth = calendar.get(Calendar.MONTH); //스피닝 캘린더로 이번 월(month) 받기
                mDay = calendar.get(Calendar.DAY_OF_MONTH); //스피닝 캘린더로 오늘 일(day) 받기
            }
        }
    }
}

```

```

        }

    }

} catch (Exception ex) {
    // 스피닝 캘린더(layout풀더에서 getdatepopup xml 파일 보시면 Datepicker라고 써있는데 종류가
    스피닝 캘린더입니다.)
    Calendar calendar = new GregorianCalendar();
    mYear = calendar.get(Calendar.YEAR); //스피닝 캘린더로 요번 년도값(year) 받기
    mMonth = calendar.get(Calendar.MONTH); //스피닝 캘린더로 이번 월(month) 받기
    mDay = calendar.get(Calendar.DAY_OF_MONTH); //스피닝 캘린더로 오늘 일(day) 받기
}

DatePicker datePicker = (DatePicker) findViewById(R.id.DatePicker); // xml의 datepicker를 자바에서
사용하기위해 캐스팅
datePicker.init(mYear, mMonth, mDay, mOnDateChangedListener); // 아래 있음

}

public void mOnClick(View v) throws Exception { // 클릭시
    mMonth += 1;
    date = mYear + "/" + expressMonth(mMonth) + "/" + expressDate(mDay);
    switch (screenview) {
        case "mainfrom": {

            Date txtOriginalFrom = format.parse(txt_From.getText().toString());
            assert txtOriginalFrom != null;

            //txt_From에 선택한것 입력 그리고 PreferenceManager로 프로그램 재시작시에도 자신이
            선택해놓았던 날짜를 기억하기위해 같은 String 저장
            txt_From.setText(date);
            mMonth -= 1;

            //받아온 txt_from과 txt_to의 임의의 날짜 String을 Date형식으로 바꾸어주기
            txtFrom = format.parse(txt_From.getText().toString()); //이름은 캐스팅한 아이템과 동일
            txtTo = format.parse(txt_to.getText().toString()); //이름은 캐스팅한 아이템과 동일
            assert txtFrom != null;
            assert txtTo != null;
            if (txtFrom.getTime() > txtTo.getTime()) {
                Toast.makeText(getApplicationContext(), "기간설정을 다시해보",
                Toast.LENGTH_SHORT).show();
                txt_From.setText(format.format(txtOriginalFrom));
                finish();
                break;
            }

            PreferenceManager.setString(getApplicationContext(), "maindatefrom", date);

            //만약에 txtFrom의 절대 시간 값이 txtTo의 절대 적인 시간 값 보다 크다면 계산적오류가 생기므로
            종료해준다.

            //임금계산 함수
            CalculateWage();
            break;
        }
        case "rwdateto": {
    }
}

```

```

final TextView txt_dateTo = ((registerWork)
registerWork.context).findViewById(R.id.txt_rwDateTo);
Date gdate = format.parse(date);
Date nextDate = format1.parse(NEXTDATE);
Date currentDate = format1.parse(CURRENTDATE);
assert gdate != null;
assert currentDate != null;
assert nextDate != null;
String changedFormatDate = format1.format(gdate);
if (gdate.getTime() >= currentDate.getTime() && gdate.getTime() <= nextDate.getTime()) {
    txt_dateTo.setText(changedFormatDate);
    mMonth -= 1;
    finish();
} else {
    Toast.makeText(getApplicationContext(), "현재날짜와 현재날짜의 다음날 까지만 입력가능해.", Toast.LENGTH_LONG).show();
}
break;
}

case "mainto": {
    Date txtOriginalTo = format.parse(txt_to.getText().toString());
    assert txtOriginalTo != null;

    //txt_to에 선택한것 입력 그리고 PreferenceManager로 프로그램 재시작시에도 자신이
    선택해놓았던 날짜를 기억하기위해 같은 String 저장
    txt_to.setText(date);
    mMonth -= 1; // 날짜를 계산할때 달은 현재 달보다 한달 더 작게 나타나서 위의 코드에서 +1
    해주었던 것을 다시 -1하여 원래대로 되돌려준다.

    //받아온 txt_from과 txt_to의 임의의 날짜 String을 Date형식으로 바꾸어주기
    txtFrom = format.parse(txt_From.getText().toString()); //이름은 캐스팅한 아이템과 동일
    txtTo = format.parse(txt_to.getText().toString()); //이름은 캐스팅한 아이템과 동일
    assert txtFrom != null;
    assert txtTo != null;
    if (txtFrom.getTime() > txtTo.getTime()) {
        Toast.makeText(getApplicationContext(), "기간설정을 다시해봐",
Toast.LENGTH_SHORT).show();
        txt_to.setText(format.format(txtOriginalTo));
        finish();
        break;
    }
    myapplication.setDate_To(date);
    PreferenceManager.setString(getApplicationContext(), "maindateto", date);

    //만약에 txtFrom의 절대 시간 값이 txtTo의 절대 적인 시간 값 보다 크다면 계산적오류가 생기므로
    종료해준다.

    //임금계산 함수
    CalculateWage();
    break;

}
}
}

```

```

// 선택한 datepicker 값이 바뀔경우 바뀔때마다 새로고침 해준다.
DatePicker.OnDateChangedListener mOnDateChangedListener = (datePicker, yy, mm, dd) -> { // 바뀐
값을 변수에 저장
    mYear = yy;
    mMonth = mm;
    mDay = dd;
};

private String expressMonth(int month) {
    String smonth = Integer.toString(month);
    if (month < 10) {
        smonth = "0" + smonth;
    }
    return smonth;
}

private String expressDate(int date) {
    String sdate = Integer.toString(date);
    if (date < 10) {
        sdate = "0" + sdate;
    }
    return sdate;
}

private long subtractDate(String dateFrom, String dateTo) {
    try {
        SimpleDateFormat format = new SimpleDateFormat("yyyy/MM/dd", Locale.KOREA);

        Date fromDate = format.parse(dateFrom);
        Date toDate = format.parse(dateTo);

        assert fromDate != null;
        assert toDate != null;
        long calDate = toDate.getTime() - fromDate.getTime();

        long calDateDays = calDate / (24 * 60 * 60 * 1000);

        calDateDays = Math.abs(calDateDays);

        System.out.println("두 날짜의 날짜 차이: " + calDateDays + "\n fromDate : " + fromDate.toString() +
" toDate : " + toDate.toString() +
" calDate : " + calDate + "\n 언제부터 : " + dateFrom + " 언제까지 : " + dateTo);

        return calDateDays + 1;
    } catch (ParseException ignored) {
        return 0;
    }
}

private Cursor getregisterCursor() {
    registerWorkDBHelper db = registerWorkDBHelper.getInstance(this);
    return db.getReadableDatabase().query(
        registerWorkContract.registerWorkEntry.TABLE_NAME, null, null, null, null, null,
registerWorkContract.registerWorkEntry._ID);
}

```

```

}

private Cursor getTotalCursor() {
    totalDB db = totalDB.getInstance(this);
    return db.getReadableDatabase().query(
        DataBases.CreatDB._TABLENAME0, null, null, null, null, null, null, DataBases.CreatDB._ID);
}

private double subHours(String dateFrom, String dateTo, String hoursFrom, String hoursTo) {
    try {
        String[] organizedHoursFrom = hoursFrom.split(" ");
        String[] organizedHoursTo = hoursTo.split(" ");

        if (organizedHoursFrom[0].equals("오후")) {
            organizedHoursFrom[1] = Integer.toString(Integer.parseInt(organizedHoursFrom[1]) + 12);
        }

        if (organizedHoursTo[0].equals("오후")) {
            organizedHoursTo[1] = Integer.toString(Integer.parseInt(organizedHoursTo[1]) + 12);
        }

        if (Integer.parseInt(organizedHoursFrom[1]) < 10) {
            organizedHoursFrom[1] = "0" + organizedHoursFrom[1];
        }

        if (Integer.parseInt(organizedHoursTo[1]) < 10) {
            organizedHoursTo[1] = "0" + organizedHoursTo[1];
        }

        hoursFrom = dateFrom + " " + organizedHoursFrom[1] + ":" + organizedHoursFrom[3];
        hoursTo = dateTo + " " + organizedHoursTo[1] + ":" + organizedHoursTo[3];

        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm", Locale.KOREA);

        Date fromDate = format.parse(hoursFrom);
        Date toDate = format.parse(hoursTo);

        assert fromDate != null;
        assert toDate != null;
        double calHours = toDate.getTime() - fromDate.getTime();

        double calDateDays = calHours / (60 * 60 * 1000);

        calDateDays = Math.abs(calDateDays);

        System.out.println("두 날짜의 날짜 차이: " + calDateDays + "\n fromDate : " + fromDate.toString() +
            " toDate : " + toDate.toString() +
            " calDate : " + calHours + "\n 언제부터 : " + hoursFrom + " 언제까지 : " + hoursTo);

        return calDateDays + 1;
    } catch (ParseException ignored) {
        return 0;
    }
}

```

```

private Date getFromHours(String date, String hoursFrom) {
    try {
        String[] organizedHoursFrom = hoursFrom.split(" ");
        if (organizedHoursFrom[0].equals("오후")) {
            organizedHoursFrom[1] = Integer.toString(Integer.parseInt(organizedHoursFrom[1]) + 12);
        }

        if (Integer.parseInt(organizedHoursFrom[1]) < 10) {
            organizedHoursFrom[1] = "0" + organizedHoursFrom[1];
        }

        hoursFrom = date + " " + organizedHoursFrom[1] + ":" + organizedHoursFrom[3];
        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm", Locale.KOREA);
        return format.parse(hoursFrom);
    } catch (ParseException e) {
        e.printStackTrace();
        return null;
    }
}

private Date getToHours(String date, String hoursTo) {
    try {
        String[] organizedHoursTo = hoursTo.split(" ");
        if (organizedHoursTo[0].equals("오후")) {
            organizedHoursTo[1] = Integer.toString(Integer.parseInt(organizedHoursTo[1]) + 12);
        }

        if (Integer.parseInt(organizedHoursTo[1]) < 10) {
            organizedHoursTo[1] = "0" + organizedHoursTo[1];
        }

        hoursTo = date + " " + organizedHoursTo[1] + ":" + organizedHoursTo[3];
        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm", Locale.KOREA);
        return format.parse(hoursTo);
    } catch (ParseException e) {
        e.printStackTrace();
        return null;
    }
}

private int getWeekOfYear(String date) {
    Calendar calendar = Calendar.getInstance();
    String[] dates = date.split("-");
    int year = Integer.parseInt(dates[0]);
    int month = Integer.parseInt(dates[1]);
    int day = Integer.parseInt(dates[2]);
    calendar.set(year, month - 1, day);
}

```

```

        return calendar.get(Calendar.WEEK_OF_MONTH);
    }

public String getDateDay(String date, String dateType, Locale locale) throws Exception {

    String day = "";

    SimpleDateFormat dateFormat = new SimpleDateFormat(dateType, locale);
    Date nDate = dateFormat.parse(date);

    Calendar cal = Calendar.getInstance();
    assert nDate != null;
    cal.setTime(nDate);

    int dayNum = cal.get(Calendar.DAY_OF_WEEK);

    switch (dayNum) {
        case 1:
            day = "sun";
            break;
        case 2:
            day = "mon";
            break;
        case 3:
            day = "tue";
            break;
        case 4:
            day = "wed";
            break;
        case 5:
            day = "thu";
            break;
        case 6:
            day = "fri";
            break;
        case 7:
            day = "sat";
            break;
    }

    return day;
}

public void CalculateWage() {

    try {

        // 날짜를 뺄셈하여 기간을 저장하고 전역변수에 저장
        long subtractDate = subtractDate(txt_From.getText().toString(), myapplication.getDate_To());

        //totalDB,registerDB의 모든 정보를 가리킬수 있는 Cursor를 만들어준다.(이름은 실제 DB이름과 동일)
        Cursor totalDBCursor = getTotalCursor();
        Cursor registerDBCursor = getregisterCursor();
    }
}

```

```

//메인 버튼의 알바이름으로 일치하는 totalDB의 알바이름이 있는 Row의 모든 Column 정보를
가져온다.
System.out.println("값 : " + subtractDate);

for (int i = 0; i < totalDBCursor.getCount(); i++) // n^2
{
    if (i > 0 && !(totalDBCursor.getString(1).equals(btn_choose.getText().toString()))) {
        i = i - 1;
    }
    totalDBCursor.moveToPosition(count);
    if (count < totalDBCursor.getCount()) {
        count = count + 1;
    } else {
        break;
    }
    System.out.println(totalDBCursor.getString(1) + " , " + btn_choose.getText().toString());
    if (totalDBCursor.getString(1).equals(btn_choose.getText().toString())) {
        for (int j = 0; j < 8; j++) {
            totalArray[i][j] = totalDBCursor.getString(j);
            System.out.println("total : " + totalArray[i][j]);
        }
    }
}

//식사시간 소수점 단위로 바꾸기
double mealtime = Double.parseDouble(totalArray[0][7]) / 60;
System.out.println("식사시간 : " + mealtime);

```

```

//메인 버튼의 알바이름으로 일치하는 registerWork의 알바이름이 있는 Row의 모든 Column 정보를
가져온다.
//fromDate는 메인의 txt_from에서 가져온 String을 Date
//count는 재사용 위해 다시 한번 초기화
count = 0;
for (int i = 0; i < registerDBCursor.getCount(); i++) {

    //registerDB의 날짜 데이터를 하나씩 체크한다.
    registerDBCursor.moveToPosition(i);
    Date registerDate = format1.parse(registerDBCursor.getString(1)); //1번은 날짜정보
    assert registerDate != null;
    System.out.println("registerDate = " + registerDate.getTime() + " txtFrom = " + txtFrom.getTime()
+ " txtTo = " + txtTo.getTime());
    //registerDB의 이름 정보와 메인의 btn_choosepartime 버튼의 텍스트와 같고
    //registerDate에 그날 알바정보를 등록한 시간이 txtFrom시간(보고싶은 알바비 기간의
    시작날짜)보다 크거나 같고 또 txtTo시간(보고싶은 알바비 기간의 끝날짜)보다 작거나 같다면
    //그 데이터를 registerArray에 집어 넣어준다.
    if (registerDBCursor.getString(2).equals(btn_choose.getText().toString()) &&
(registerDate.getTime() >= txtFrom.getTime() && registerDate.getTime() <= txtTo.getTime())) {
        for (int j = 0; j < 9; j++) {
            registerArray[count][j] = registerDBCursor.getString(j);
        }
        count++;
    }
}

//registerArray 내용 확인용(지우지 마시오:버그패치용 이기도함)
for (int i = 0; i < count; i++) {
    for (int j = 0; j < 9; j++) {

```

```

        System.out.println("registerArray = " + registerArray[i][j]);
    }
}

//registerArray의 뒤쪽 박죽 일수 있는 필요한! registerArray정보를 날짜 순서대로 버블정렬한다. 1번
= 날짜,3번 = 시작시간,4번 = 퇴근시간,8번 = 퇴근날짜 끼리만 버블 정렬
for (int i = 0; i < count; i++) {
    for (int j = i + 1; j < count; j++) {
        System.out.println("registerArray Date = " + registerArray[i][1]);
        Date date = format1.parse(registerArray[i][1]);
        if (registerArray[j][1] != null) {
            Date date1 = format1.parse(registerArray[j][1]);
            assert date1 != null;
            assert date != null;
            if (date.getTime() > date1.getTime()) {
                String tmp = registerArray[i][1];
                registerArray[i][1] = registerArray[j][1];
                registerArray[j][1] = tmp;
                String tmp1 = registerArray[i][3];
                registerArray[i][3] = registerArray[j][3];
                registerArray[j][3] = tmp1;
                String tmp2 = registerArray[i][4];
                registerArray[i][4] = registerArray[j][4];
                registerArray[j][4] = tmp2;
                String tmp3 = registerArray[i][8];
                registerArray[i][8] = registerArray[j][8];
                registerArray[j][8] = tmp3;
            }
        } else {
            break;
        }
    }
}

//registerArray의 1 index = 날짜,3 index = 시작시간,4 index = 퇴근시간,8 index = 퇴근날짜 확인용
(디버그용 지우지 마세요!)
for (int i = 0; i < count; i++) {
    System.out.println("registerArray = (" + registerArray[i][1] + " , " + registerArray[i][3] + " , " +
registerArray[i][4] + " , " + registerArray[i][8] + ")");
}

//count = registerArray에서 위의 모든 조건에 합당한 데이터 row의 갯수(즉 원하는 날짜사이의 필요한
데이터들)
//workinghours에 registerArray의 버블 정렬한것의 시간데이터만 순서대로 넣어주고
hoursSubtractions에 to와 from의 시간차를 subHours()로 구해서 넣어준다.
for (int i = 0; i < count; i++) {
    workingHours.add(new String[]{registerArray[i][1], registerArray[i][3], registerArray[i][4]});
    System.out.println(workingHours.get(i)[0]);
    String from = workingHours.get(i)[1];
    String to = workingHours.get(i)[2];
    hoursSubtractions.add(new String[]{registerArray[i][1],
String.valueOf(subHours(registerArray[i][1], registerArray[i][8], from, to) - 1 - mealtime)}));
}

```

```

//위에서 구한 hoursSubtractions 이차원 배열을 Locat에 표시해준다.
for (int i = 0; i < count; i++) {
    System.out.println("hoursSubtractionsArray = (" + hoursSubtractions.get(i)[0] + " , " +
hoursSubtractions.get(i)[1]);
}

//기본급으로 계산한시간 (연장/야간 적용 없이) //휴일수당,주휴수당 포함
//그 날 하루급여를 기준으로 계산
// 1 기본급
for (int i = 0; i < count; i++) {
    totalWorkingHours = Double.parseDouble(hoursSubtractions.get(i)[1]);
    System.out.println("하루 일한시간 : " + i + " = " + totalWorkingHours);
    oneDayWage[i] = totalWorkingHours * Double.parseDouble(totalArray[0][2]);

    //연장무수당 하루 통상근무 시간인 8시간을 초과했을 때 또는 주급으로 따진다면 주 40시간 초과
    했을 때 적용(1.5배)
    if (totalWorkingHours > 8) {
        double extendedHours = totalWorkingHours - 8;
        double extendedWage = extendedHours * (Double.parseDouble(totalArray[0][2]) * 0.5);
        oneDayWage[i] = oneDayWage[i] + extendedWage;
    }

    //야간 근무수당 밤 10시부터 오전 6시까지 일을 했을경우 그 사이의 일한 시간에 대하여 1.5를
    받는다. 연장수당에 가산하여 받는다.(연장+야간 겹치기 가능)
    Date tmpFromHours = timeDateFormat.parse(registerArray[i][1] + " 22:00"); //근무당일 밤 10시
    Date today = format1.parse(registerArray[i][1]); //오늘 날짜
    // 내일 날짜 구하기
    assert today != null;
    long tod = today.getTime() + 86400000; //근무당일 날짜 + 86400000(밀리초로 구한하루)
    Date tomorrow = new Date(tod); //근무일다음날 날짜
    String tom = format1.format(tomorrow); // 근무일다음날 0시 0분 0초를 format1을 String인
    yyyy-MM-dd HH:mm:ss 날짜형식으로 만들어 tom(tomorrow의 약자)에 넣어줌
    Date tmpToHours = timeDateFormat.parse(tom + " 06:00"); // tmpTohours = 근무일다음날 6시
    Date tmpCurrentFromHours = timeDateFormat.parse(registerArray[i][1] + " 06:00"); // 근무당일
    6시
    Date fromHours = getFromHours(registerArray[i][1], registerArray[i][3]); // registerwork의
    근무시작시간
    Date ToHours = getToHours(registerArray[i][8], registerArray[i][4]); // registerwork의 근무종료시간
    assert fromHours != null;
    assert tmpToHours != null;
    assert tmpFromHours != null;
    assert ToHours != null;
    assert tmpCurrentFromHours != null;
    //야간수당 조건을 위한 시간의 기준들의 절대적 값을 Logcat 으로 확인(디버그용 지우지 마세요)
    System.out.println(fromHours.getTime() + " , " + tmpFromHours.getTime() + " , " +
    ToHours.getTime() + " , " + tmpToHours.getTime());
    System.out.println(fromHours.getTime() + " , " + tmpCurrentFromHours.getTime() + " , " +
    ToHours.getTime() + " , " + tmpCurrentFromHours.getTime());
    System.out.println(fromHours.getTime() + " , " + tmpCurrentFromHours.getTime() + " , " +
    ToHours.getTime() + " , " + tmpCurrentFromHours.getTime());

    //야간 근무 수당 적용을 위한 조건들
    //조건 1 : 근무시작시간 >= 근무당일 22시 && 근무시작시간 <= 근무일다음날 6시&& 근무종료시간
    <= 근무일다음날 6시 && 근무종료시간 >= 근무당일 22시
    //조건 2 : 근무시작시간 >= 근무당일 6시 && 근무시작시간 <= 근무당일 22시 && 근무종료시간 <=
    근무일다음날 6시 && 근무종료시간 >= 근무당일 22시
    //조건 3 : 근무시작시간 <= 근무당일 6시 && 근무종료시간 <= 근무당일 6시 && 근무종료시간 >

```

```

근무시작시간
    //조건 4 : 근무시작시간 <= 근무당일 6시 && 근무종료시간 >= 근무당일 6시 && 근무종료시간 <=
    근무당일 22시
        if ((fromHours.getTime() >= tmpFromHours.getTime() && fromHours.getTime() <=
            tmpToHours.getTime() && ToHours.getTime() <= tmpToHours.getTime() && ToHours.getTime() >=
            tmpFromHours.getTime())) {
            double nightWorkingWage = totalWorkingHours * (Double.parseDouble(totalArray[0][2]) * 0.5);
            System.out.println("hours" + totalWorkingHours);
            oneDayWage[i] = oneDayWage[i] + nightWorkingWage;
        } else if (fromHours.getTime() >= tmpCurrentFromHours.getTime() && fromHours.getTime() <=
            tmpFromHours.getTime() && ToHours.getTime() <= tmpToHours.getTime() && ToHours.getTime() >=
            tmpFromHours.getTime()) {
            double hours = Double.parseDouble(String.valueOf(ToHours.getTime() -
            tmpFromHours.getTime())) / (1000 * 60 * 60);
            System.out.println("hours1 : " + i + " = " + hours);
            double nightWorkingWage = hours * (Double.parseDouble(totalArray[0][2]) * 0.5);
            oneDayWage[i] = oneDayWage[i] + nightWorkingWage;
        } else if (fromHours.getTime() <= tmpCurrentFromHours.getTime() && ToHours.getTime() <=
            tmpCurrentFromHours.getTime() && ToHours.getTime() > fromHours.getTime()) {
            double hours = Double.parseDouble(String.valueOf(ToHours.getTime() -
            fromHours.getTime())) / (1000 * 60 * 60);
            double nightWorkingWage = hours * (Double.parseDouble(totalArray[0][2]) * 0.5);
            System.out.println("hours2" + hours);
            oneDayWage[i] = oneDayWage[i] + nightWorkingWage;
        } else if (fromHours.getTime() <= tmpCurrentFromHours.getTime() && ToHours.getTime() >=
            tmpCurrentFromHours.getTime() && ToHours.getTime() <= tmpFromHours.getTime()) {
            double hours = Double.parseDouble(String.valueOf(tmpCurrentFromHours.getTime() -
            fromHours.getTime())) / (1000 * 60 * 60);
            double nightWorkingWage = hours * (Double.parseDouble(totalArray[0][2]) * 0.5);
            System.out.println("hours3" + hours);
            oneDayWage[i] = oneDayWage[i] + nightWorkingWage;
        }
    }

    //주말수당 검사 및 주말수당 적용
    //배열날짜(hoursSubtractions.get(i)[0]) == sat or sun 이라면 (그날의 날짜 == 토요일 또는
    일요일이라면)
    if (getDateDay(hoursSubtractions.get(i)[0], "yyyy-MM-dd", Locale.KOREA).equals("sat") ||
        getDateDay(hoursSubtractions.get(i)[0], "yyyy-MM-dd", Locale.KOREA).equals("sun")) {
        double tmpBreakWage = totalWorkingHours * (Double.parseDouble(totalArray[0][2]) * 0.5);
        oneDayWage[i] = oneDayWage[i] + tmpBreakWage;
    }

    System.out.println("하루 기본급 : " + i + " = " + oneDayWage[i]); //기본시급과 연장,주휴,야간을
    포함한 하루 기본급 Locat에 표시 디버그용 지우지마세요
}

//주휴 수당
double[] weekWorkingHours = new double[5];
try {
    for (int i = 0; i < count; i++) {
        //저장된 날짜가 그달의 몇주째인지 계산해서 weekNum(week number)에 넣어준다.
        int weekNum = getWeekOfYear(hoursSubtractions.get(i)[0]);
        System.out.println("weekNum = " + weekNum + ", " + hoursSubtractions.get(i)[0]); //디버그용
        지우지마세요
        String DateDay = getDateDay(hoursSubtractions.get(i)[0], "yyyy-MM-dd", Locale.KOREA);
        //근무당일날짜의 요일 getDateDay가 날짜를 그 날짜의 요일로 바꿔줌
        //받은 날짜가 1주차 날짜라면
    }
}

```

```

if (weekNum == 1) {
    String[] dayOfWeeks = totalArray[0][3].split(","); // totalArray에 기본적으로 근무를 하기로
정해진 요일들을 dayOfWeeks에 배열로 나누어 넣음
    System.out.println("dayOfWeeks1" + dayOfWeeks.length); //디버그용 지우지마세요
    for (int j = 0; j < 7; j++) {
        if (dayOfWeeks[j].equals(DateDay)) { //근무하기로 정해진 요일 == 근무당일날짜의 요일
            firstWeek += 1; // firstWeek 첫째주의 근무나간 횟수를 의미하는 int변수 +1
            weekWorkingHours[0] = weekWorkingHours[0] +
Double.parseDouble(hoursSubtractions.get(i)[1]); // 그주의 일한시간 = 그주의 일한시간 +
근무당일일한시간
            System.out.println("firstweek = " + weekWorkingHours[0]);//디버그용 지우지마세요
        }
    }
}
//받은 날짜가 2주차 날짜라면 (과정은 위와 동일)
else if (weekNum == 2) {
    String[] dayOfWeeks = totalArray[0][3].split(",");
    System.out.println("dayOfWeeks2" + dayOfWeeks.length); //디버그용 지우지마세요
    for (int j = 0; j < 7; j++) {
        if (dayOfWeeks[j].equals(DateDay)) {
            secondWeek += 1;
            weekWorkingHours[1] = weekWorkingHours[1] +
Double.parseDouble(hoursSubtractions.get(i)[1]);
            System.out.println("secondweek = " + weekWorkingHours[1]);
        }
    }
}
//받은 날짜가 3주차 날짜라면(과정은 위와 동일)
else if (weekNum == 3) {
    String[] dayOfWeeks = totalArray[0][3].split(",");
    System.out.println("dayOfWeeks3" + dayOfWeeks.length); //디버그용 지우지마세요
    for (int j = 0; j < 7; j++) {
        if (dayOfWeeks[j].equals(DateDay)) {
            thirdWeek += 1;
            weekWorkingHours[2] = weekWorkingHours[2] +
Double.parseDouble(hoursSubtractions.get(i)[1]);
            System.out.println("thirdweek = " + weekWorkingHours[2]); //디버그용 지우지마세요
        }
    }
}
//받은 날짜가 4주차 날짜라면(과정은 위와 동일)
else if (weekNum == 4) {
    String[] dayOfWeeks = totalArray[0][3].split(",");
    System.out.println("dayOfWeeks4" + dayOfWeeks.length); //디버그용 지우지마세요
    for (int j = 0; j < 7; j++) {
        if (dayOfWeeks[j].equals(DateDay)) {
            forthWeek += 1;
            weekWorkingHours[3] = weekWorkingHours[3] +
Double.parseDouble(hoursSubtractions.get(i)[1]);
            System.out.println("forthweek = " + weekWorkingHours[3]); //디버그용 지우지마세요
        }
    }
}
//받은 날짜가 5주차 날짜라면(과정은 위와 동일)
else if (weekNum == 5) {
    String[] dayOfWeeks = totalArray[0][3].split(",");
    System.out.println("dayOfWeeks5" + dayOfWeeks.length); //디버그용 지우지마세요
    for (int j = 0; j < 7; j++) {
        if (dayOfWeeks[j].equals(DateDay)) {
            fifthWeek += 1;
            weekWorkingHours[4] = weekWorkingHours[4] +
Double.parseDouble(hoursSubtractions.get(i)[1]);
    }
}

```

```

        System.out.println("fifthweek = " + weekWorkingHours[4]); //디버그용 지우지마세요
    }
}
}
}
//위의 try문이 실패할경우 즉 저장된 데이터가 없을경우
catch (Exception e) {
    e.printStackTrace();
}

//



String[] dayOfWeeks = totalArray[0][3].split(",");
int dayofweekcount = 0;
double value;
for (String dayOfWeek : dayOfWeeks) {
    if (!dayOfWeek.equals("null")) {
        dayofweekcount += 1;
    }
}
//만약 그 grid의 개수가 totalarray의 요일의 개수와 일치하고 그 주의 총 근무시간이 15시간 이상이라면 주휴수당을 받아야한다.
// (1주일 총근로시간/40) * totalArray를 통해서 구한 근무시간 * 시급
if (firstWeek == dayofweekcount && weekWorkingHours[0] >= 15) {
    if (weekWorkingHours[0] >= 40) {
        weekWorkingHours[0] = 40;
    }
    value = (weekWorkingHours[0] / 40) * 8 * Double.parseDouble(totalArray[0][2]);
    extraMoney[0] = value;
    System.out.println("extraMoney 0 " + extraMoney[0]);
}
if (secondWeek == dayofweekcount && weekWorkingHours[1] >= 15) {
    if (weekWorkingHours[1] >= 40) {
        weekWorkingHours[1] = 40;
    }
    value = (weekWorkingHours[1] / 40) * 8 * Integer.parseInt(totalArray[0][2]);
    extraMoney[1] = value;
    System.out.println("extraMoney 1 " + extraMoney[1]);
}
if (thirdWeek == dayofweekcount && weekWorkingHours[2] >= 15) {
    if (weekWorkingHours[2] >= 40) {
        weekWorkingHours[2] = 40;
    }
    value = (weekWorkingHours[2] / 40) * 8 * Integer.parseInt(totalArray[0][2]);
    extraMoney[2] = value;
    System.out.println("extraMoney 2 " + extraMoney[2]);
}
if (forthWeek == dayofweekcount && weekWorkingHours[3] >= 15) {
    if (weekWorkingHours[3] >= 40) {
        weekWorkingHours[3] = 40;
    }
    value = (weekWorkingHours[3] / 40) * 8 * Integer.parseInt(totalArray[0][2]);
    extraMoney[3] = value;
    System.out.println("extraMoney 3 " + extraMoney[3]);
}
if (fifthWeek == dayofweekcount && weekWorkingHours[4] >= 15) {
    if (weekWorkingHours[4] >= 40) {
        weekWorkingHours[4] = 40;
    }
}

```

```

        value = (weekWorkingHours[4] / 40) * 8 * Integer.parseInt(totalArray[0][2]);
        extraMoney[4] = value;
        System.out.println("extraMoney 4 " + extraMoney[4]);
    }
    double extramoney = 0;
    try {
        for (int i = 0; i < 5; i++) {
            try {
                extramoney += extraMoney[i];
            } catch (Exception ex) {
                ex.printStackTrace();
            }
            System.out.println("extramoney " + extramoney);
        }
    } catch (Exception e) {
        System.out.println("주휴수당은 없습니다.");
    }

    //extramoney = 총 주휴수당 여기다 모든 oneDayWage를 더하면 그 기간동안의 법정 최소한의
    급여가 된다.
    for (int i = 0; i < count; i++) {
        totalWage += oneDayWage[i];
        System.out.println(i + " : " + oneDayWage[i]);
    }
    totalWage = totalWage + extramoney;

    EditText mainEdit = ((MainActivity) MainActivity.context_main).findViewById(R.id.txt_monthlyPay);
    PreferenceManager.setString(getApplicationContext(), "suggestedSavedWage",
String.format(Locale.KOREA, "%d 원", (int) totalWage));

    if(PreferenceManager.getString(getApplicationContext(),"probationPeriodChecked").equals("true")){
        String tmpString = ((int)totalWage/100*90) + " 원";
        mainEdit.setText(tmpString);
    }
    else
    {
        mainEdit.setText(String.format(Locale.KOREA, "%d 원", (int) totalWage));
    }

    finish();

} catch (Exception ex) {
    final TextView txt_to = ((MainActivity) MainActivity.context_main).findViewById(R.id.txt_to);
    txt_to.setText(date);
    PreferenceManager.setString(getApplicationContext(), "maindateto", date);
    mMonth -= 1;
    myapplication.setDate_To(date);
    finish();
}
}
}

```

intro.java

```
package com.alba.lionproject1;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.TextView;

public class Intro extends AppCompatActivity {

    TextView txt_extra_p;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_intro);

        txt_extra_p = (TextView) findViewById(R.id.txt_extra_p);

        Animation anim = AnimationUtils.loadAnimation
            (getApplicationContext(), // 현재화면의 제어권자
             R.anim.translate_anim); // 애니메이션 설정 파일
        txt_extra_p.startAnimation(anim);

        //인트로 화면에서 화면 지나갈때 지나가는 시간을 지연시켜주기
        Handler handler = new Handler();
        handler.postDelayed(() -> {
            Intent intent = new Intent(Intro.this, MainActivity.class);
            startActivity(intent);
            finish();
        }, 4000); // 2초 지연
    }
}
```

MainActivity.java

```
package com.alba.lionproject1;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import android.app.Activity;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import android.widget ScrollView;
import android.widget.TextView;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Calendar;
import java.util.Date;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {
    ArrayList<String[]> hoursSubtractions = new ArrayList<>();
    ArrayList<String[]> workingHours = new ArrayList<>();
    double[] oneDayWage = new double[100000];
    String[][] totalArray = new String[100][8];
    String[][] registerArray = new String[100000][9];
    double totalWage = 0;
    double totalWorkingHours = 0;
    double[] extraMoney = new double[5];
    int firstWeek = 0;
    int secondWeek = 0;
    int thirdWeek = 0;
    int forthWeek = 0;
    int fifthWeek = 0;
    int count = 0;
    Myapplication myapplication;
    SimpleDateFormat format1 = new SimpleDateFormat("yyyy-MM-dd", Locale.KOREA);
    SimpleDateFormat format = new SimpleDateFormat("yyyy/MM/dd", Locale.KOREA);
    SimpleDateFormat timeDateFormat = new SimpleDateFormat("yyyy-MM-dd HH:mm", Locale.KOREA);
    Date txtFrom;
    Date txtTo;
    TextView txt_mfrom;
    TextView txt_mto;
    public static Activity main_activity;
    public static Context context_main; // context 변수 선언
    Button btn_ChooseParttime;
    Button btn_ImgFilePath;
    Button btn_EvidencePath;
    String suggestedSavedWage = "";
    EditText showWage;
    CheckBox cb_probationPeriod;
    ImageButton btn_ProbationPeriod;
```

```

@RequiresApi(api = Build.VERSION_CODES.R)
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    context_main = this;
    main_activity = MainActivity.this;
    myapplication = (Myapplication) getApplication();

    // 국가 설정(한국으로) - 처음에 커스텀 달력이 아니라 기본 달력 사용 했을 때 국가설정해야 언어 바뀌어서 그렇게 했어
    Locale.setDefault(Locale.KOREA);
    this.setContentView(R.layout.activity_main);

    Date currentTime = Calendar.getInstance().getTime();
    SimpleDateFormat dateFormat = new SimpleDateFormat("dd", Locale.KOREA);
    SimpleDateFormat monthFormat = new SimpleDateFormat("MM", Locale.KOREA);
    SimpleDateFormat yearFormat = new SimpleDateFormat("yyyy", Locale.KOREA);

    String year = yearFormat.format(currentTime);
    String month = monthFormat.format(currentTime);
    String day = dateFormat.format(currentTime);
    String date = year + "/" + expressMonth(Integer.parseInt(month)) + "/" +
expressDate(Integer.parseInt(day));

    txt_mfrom = findViewById(R.id.txt_from);
    txt_mto = findViewById(R.id.txt_to);
    btn_ChooseParttime = findViewById(R.id.btn_ChooseParttime);
    btn_ImgFilePath = findViewById(R.id.btn_ImgFilePath);
    btn_EvidencePath = findViewById(R.id.btn_EvidenceCsv);
    showWage = findViewById(R.id.txt_monthlyPay);
    cb_probationPeriod = findViewById(R.id.cb_susoup);
    LinearLayout linearLayout = findViewById(R.id.sclinear);
    ScrollView scrollView = findViewById(R.id.scroll);
    btn_ProbationPeriod = findViewById(R.id.btn_helpProbationPeriod);

    ScrollView lineLayout = (ScrollView) linearLayout.getParent();
    int x = lineLayout.getLeft();
    int y = lineLayout.getTop();
    scrollView.smoothScrollTo(x, y);

    try {
        myapplication.setMainDateFrom(PreferenceManager.getString(getApplicationContext(),
"maindatefrom"));
        myapplication.setMainDateTo(PreferenceManager.getString(getApplicationContext(),
"maindateto"));
    } catch (Exception ex) {
        ex.printStackTrace();
    }

    txt_mfrom.setText(myapplication.getMainDateFrom());
    txt_mto.setText(myapplication.getMainDateTo());
    if (txt_mfrom.getText().toString().equals("")) {
        txt_mfrom.setText(date);
    }
}

```

```

if (txt_mto.getText().toString().equals("")) {
    txt_mto.setText(date);
}

try {
    myapplication.setNameOfAlba(PreferenceManager.getString(context_main, "albaname"));
} catch (Exception ex) {
    ex.printStackTrace();
}
btn_ChooseParttime.setText(myapplication.getNameOfAlba());
if (btn_ChooseParttime.getText().toString().equals("")) {
    btn_ChooseParttime.setText("알바 선택해");
}

try {
    txtFrom = format.parse(txt_mfrom.getText().toString());
    txtTo = format.parse(txt_mto.getText().toString());
    CalculateWage();
} catch (ParseException ex) {
    showWage.setText("0 원");
}
try{
    if(PreferenceManager.getString(getApplicationContext(),"probationPeriodChecked").equals("true"))
{
    cb_probationPeriod.setChecked(true);
    String[] tmpTotalWage = showWage.getText().toString().split(" ");
    System.out.println("tmpTotalWage = " + tmpTotalWage[0]);
    double probationPeriod = Double.parseDouble(tmpTotalWage[0]) / 100 * 90;
    System.out.println("probation = " + probationPeriod);
    tmpTotalWage[0] = (int)probationPeriod + " 원";
    showWage.setText(tmpTotalWage[0]);
}
} catch (Exception ex) {
    System.out.println("probation 실패");
}

cb_probationPeriod.setOnClickListener(view -> {
    if (cb_probationPeriod.isChecked())
    {
        PreferenceManager.setString(getApplicationContext(),"probationPeriodChecked", "true");
        String[] tmpTotalWage = showWage.getText().toString().split(" ");
        System.out.println("tmpTotalWage = " + tmpTotalWage[0]);
        double probationPeriod = Double.parseDouble(tmpTotalWage[0]) / 100 * 90;
        System.out.println("probation = " + probationPeriod);
        tmpTotalWage[0] = (int) probationPeriod + " 원";
        showWage.setText(tmpTotalWage[0]);
    }else{
        PreferenceManager.setString(getApplicationContext(),"probationPeriodChecked", "false");
        String[] tmpTotalWage = showWage.getText().toString().split(" ");
        double probationPeriod = Double.parseDouble(tmpTotalWage[0]) / 90 * 100;
        String tmp = (int)probationPeriod + " 원";
        showWage.setText(tmp);
    }
};

//알바비를 표시해주기 위하여 자신이 원하는 언제부터 언제 까지 를 설정하기 위해 날짜 선택 화면으로
넘어가는 버튼 from 이벤트

```

```

        Button btn_From = findViewById(R.id.btn_spFrom);
        btn_From.setOnClickListener(view -> {
            Intent intent = new Intent(MainActivity.this, getDatePopUp.class);
            intent.putExtra("view", "mainfrom");
            startActivity(intent);
        });

    });

    // 수습기간 도움말
    btn_ProbationPeriod.setOnClickListener(view -> {
        Intent intent_probatinoperiod = new Intent(MainActivity.this, ProbationPeriod_help.class);
        startActivity(intent_probatinoperiod);
    });

    //알바비를 표시해주기 위하여 자신이 원하는 언제부터 언제 까지 를 설정하기 위해 날짜 선택 화면으로
    넘어가는 버튼 to 이벤트
    Button btn_To = findViewById(R.id.btn_spTo);
    btn_To.setOnClickListener(view -> {
        Intent intent = new Intent(MainActivity.this, getDatePopUp.class);
        intent.putExtra("view", "mainto");
        startActivity(intent);
    });

    //무슨 알바인지 추가하고 선택할수있는 버튼 (알바1) 클릭시 이벤트
    btn_ChooseParttime.setOnClickListener(view -> {
        Intent intent = new Intent(MainActivity.this, SelectParttime.class);
        startActivity(intent);
    });

    // 이미지 파일 경로를 볼 수 있는 버튼
    btn_ImgFilePath.setOnClickListener(view -> {
        Intent intent_imgfilepath = new Intent(MainActivity.this, AlertDialog_filepath.class);
        startActivity(intent_imgfilepath);
    });

    // 증거물 파일 경로를 볼 수 있는 버튼
    btn_EvidencePath.setOnClickListener(view -> {
        Intent intent_evidencepath = new Intent(MainActivity.this, AlertDialog_addxls.class);
        startActivity(intent_evidencepath);
    });
}

private String expressMonth(int month) {
    String smonth = Integer.toString(month);
    if (month < 10) {
        smonth = "0" + smonth;
    }
    return smonth;
}

private String expressDate(int date) {
    String sdate = Integer.toString(date);
    if (date < 10) {
        sdate = "0" + sdate;
    }
}

```

```

    }
    return sdate;
}

private long subtractDate(String dateFrom, String dateTo) {
    try {
        SimpleDateFormat format = new SimpleDateFormat("yyyy/MM/dd", Locale.KOREA);

        Date fromDate = format.parse(dateFrom);
        Date toDate = format.parse(dateTo);

        assert fromDate != null;
        assert toDate != null;
        long calDate = toDate.getTime() - fromDate.getTime();

        long calDateDays = calDate / (24 * 60 * 60 * 1000);

        calDateDays = Math.abs(calDateDays);

        System.out.println("두 날짜의 날짜 차이: " + calDateDays + "\n fromDate : " + fromDate.toString() +
        " toDate : " + toDate.toString() +
        " calDate : " + calDate + "\n 언제부터 : " + dateFrom + " 언제까지 : " + dateTo);

        return calDateDays + 1;
    } catch (ParseException ignored) {
        return 0;
    }
}

private Cursor getregisterCursor() {
    registerWorkDBHelper db = registerWorkDBHelper.getInstance(this);
    return db.getReadableDatabase().query(
        registerWorkContract.registerWorkEntry.TABLE_NAME, null, null, null, null, null,
        registerWorkContract.registerWorkEntry._ID);
}

private Cursor getTotalCursor() {
    totalDB db = totalDB.getInstance(this);
    return db.getReadableDatabase().query(
        DataBases.CreatDB._TABLENAME0, null, null, null, null, null, DataBases.CreatDB._ID);
}

private double subHours(String dateFrom, String dateTo, String hoursFrom, String hoursTo) {
    try {
        String[] organizedHoursFrom = hoursFrom.split(" ");
        String[] organizedHoursTo = hoursTo.split(" ");

        if (organizedHoursFrom[0].equals("오후")) {
            organizedHoursFrom[1] = Integer.toString(Integer.parseInt(organizedHoursFrom[1]) + 12);
        }

        if (organizedHoursTo[0].equals("오후")) {
            organizedHoursTo[1] = Integer.toString(Integer.parseInt(organizedHoursTo[1]) + 12);
        }

        if (Integer.parseInt(organizedHoursFrom[1]) < 10) {

```

```

        organizedHoursFrom[1] = "0" + organizedHoursFrom[1];
    }

    if (Integer.parseInt(organizedHoursTo[1]) < 10) {
        organizedHoursTo[1] = "0" + organizedHoursTo[1];
    }

hoursFrom = dateFrom + " " + organizedHoursFrom[1] + ":" + organizedHoursFrom[3];
hoursTo = dateTo + " " + organizedHoursTo[1] + ":" + organizedHoursTo[3];

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm", Locale.KOREA);

Date fromDate = format.parse(hoursFrom);
Date toDate = format.parse(hoursTo);

assert fromDate != null;
assert toDate != null;
double calHours = toDate.getTime() - fromDate.getTime();

double calDateDays = calHours / (60 * 60 * 1000);

calDateDays = Math.abs(calDateDays);

System.out.println("두 날짜의 날짜 차이: " + calDateDays + "\n fromDate : " + fromDate.toString() +
" toDate : " + toDate.toString() +
" calDate : " + calHours + "\n 언제부터 : " + hoursFrom + " 언제까지 : " + hoursTo);

return calDateDays + 1;
} catch (ParseException ignored) {
    return 0;
}
}

private Date getFromHours(String date, String hoursFrom) {
try {
    String[] organizedHoursFrom = hoursFrom.split(" ");
    if (organizedHoursFrom[0].equals("오후")) {
        organizedHoursFrom[1] = Integer.toString(Integer.parseInt(organizedHoursFrom[1]) + 12);
    }

    if (Integer.parseInt(organizedHoursFrom[1]) < 10) {
        organizedHoursFrom[1] = "0" + organizedHoursFrom[1];
    }

hoursFrom = date + " " + organizedHoursFrom[1] + ":" + organizedHoursFrom[3];

SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm", Locale.KOREA);

return format.parse(hoursFrom);

} catch (ParseException e) {

```

```

        e.printStackTrace();
        return null;
    }

}

private Date getToHours(String date, String hoursTo) {
    try {
        String[] organizedHoursTo = hoursTo.split(" ");

        if (organizedHoursTo[0].equals("오후")) {
            organizedHoursTo[1] = Integer.toString(Integer.parseInt(organizedHoursTo[1]) + 12);
        }

        if (Integer.parseInt(organizedHoursTo[1]) < 10) {
            organizedHoursTo[1] = "0" + organizedHoursTo[1];
        }

        hoursTo = date + " " + organizedHoursTo[1] + ":" + organizedHoursTo[3];

        SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm", Locale.KOREA);

        return format.parse(hoursTo);
    } catch (ParseException e) {
        e.printStackTrace();
        return null;
    }
}

private int getWeekOfYear(String date) {
    Calendar calendar = Calendar.getInstance();
    String[] dates = date.split("-");
    int year = Integer.parseInt(dates[0]);
    int month = Integer.parseInt(dates[1]);
    int day = Integer.parseInt(dates[2]);
    calendar.set(year, month - 1, day);
    return calendar.get(Calendar.WEEK_OF_MONTH);
}

public String getDateDay(String date, String dateType, Locale locale) throws Exception {

    String day = "";

    SimpleDateFormat dateFormat = new SimpleDateFormat(dateType, locale);
    Date nDate = dateFormat.parse(date);

    Calendar cal = Calendar.getInstance();
    assert nDate != null;
    cal.setTime(nDate);

    int dayNum = cal.get(Calendar.DAY_OF_WEEK);

    switch (dayNum) {
        case 1:
            day = "sun";
            break;
        case 2:
    }
}

```

```

        day = "mon";
        break;
    case 3:
        day = "tue";
        break;
    case 4:
        day = "wed";
        break;
    case 5:
        day = "thu";
        break;
    case 6:
        day = "fri";
        break;
    case 7:
        day = "sat";
        break;
    }

    return day;
}

public void CalculateWage() {
    try {

        // 날짜를 뺄셈하여 기간을 저장하고 전역변수에 저장
        long subtractDate = subtractDate(txt_mfrom.getText().toString(), txt_mto.getText().toString());

        //totalDB,registerDB의 모든 정보를 가리킬 수 있는 Cursor를 만들어준다.(이름은 실제 DB이름과 동일)
        Cursor totalDBCursor = getTotalCursor();
        Cursor registerDBCursor = getregisterCursor();

        //메인 버튼의 알바이름과 일치하는 totalDB의 알바이름이있는 Row의 모든 Column정보를
        가져온다.
        System.out.println("값 : " + subtractDate);

        for (int i = 0; i < totalDBCursor.getCount(); i++) // n^2
        {
            System.out.println("for문 입장");
            if (i > 0 && !(totalDBCursor.getString(1).equals(btn_ChooseParttime.getText().toString())))
            {
                i = i - 1;
            }
            totalDBCursor.moveToPosition(count);
            if (count < totalDBCursor.getCount())
            {
                count = count + 1;
            } else {
                break;
            }
            System.out.println(totalDBCursor.getString(1) + " , " + btn_ChOOSEPARTTIME.getText().toString());
            if (totalDBCursor.getString(1).equals(btn_ChOOSEPARTTIME.getText().toString()))
            {
                for (int j = 0; j < 8; j++)
                {
                    totalArray[i][j] = totalDBCursor.getString(j);
                    System.out.println("total : " + totalArray[i][j]);
                }
            }
        }
    }
}

```

```

}

//식사시간 소수점 단위로 바꾸기
double mealtime = Double.parseDouble(totalArray[0][7]) / 60;
System.out.println("식사시간 : " + mealtime);

//메인 버튼의 알바이름으로 일치하는 registerWork의 알바이름이 있는 Row의 모든 Column정보를
가져온다.
//fromDate는 메인의 txt_from에서 가져온 String을 Date
//count는 재사용 위해 다시한번 초기화
count = 0;
for (int i = 0; i < registerDBCursor.getCount(); i++) {

    //registerDB의 날짜 데이터를 하나씩 체크한다.
    registerDBCursor.moveToPosition(i);
    Date registerDate = format1.parse(registerDBCursor.getString(1)); //1번은 날짜정보
    assert registerDate != null;
    System.out.println("registerDate = " + registerDate.getTime() + " txtFrom = " + txtFrom.getTime()
+ " txtTo = " + txtTo.getTime());
    //registerDB의 이름 정보와 메인의 btn_chooseparttime 버튼의 텍스트와 같고
    //registerDate에 그날 알바정보를 등록한 시간이 txtFrom시간(보고싶은 알바비 기간의
    시작날짜)보다 크거나 같고 또 txtTo시간(보고싶은 알바비 기간의 끝날짜)보다 작거나 같다면
    //그 데이터를 registerArray에 집어 넣어준다.
    if (registerDBCursor.getString(2).equals(btn_ChooseParttime.getText().toString()) &&
(registerDate.getTime() >= txtFrom.getTime() && registerDate.getTime() <= txtTo.getTime())) {
        for (int j = 0; j < 9; j++) {
            registerArray[count][j] = registerDBCursor.getString(j);
        }
        count++;
    }
}

//registerArray 내용 확인용(지우지 마시오:버그패치용 이기도함)
for (int i = 0; i < count; i++) {
    for (int j = 0; j < 9; j++) {
        System.out.println("registerArray = " + registerArray[i][j]);
    }
}

//registerArray의 뒤죽박죽 일수 있는 필요한! registerArray 정보를 날짜 순서대로 버블정렬한다. 1번
= 날짜, 3번 = 시작시간, 4번 = 퇴근시간, 8번 = 퇴근날짜 끼리만 버블 정렬
for (int i = 0; i < count; i++) {
    for (int j = i + 1; j < count; j++) {
        System.out.println("registerArray Date = " + registerArray[i][1]);
        Date date = format1.parse(registerArray[i][1]);
        if (registerArray[j][1] != null) {
            Date date1 = format1.parse(registerArray[j][1]);
            assert date1 != null;
            assert date != null;
            if (date.getTime() > date1.getTime()) {
                String tmp = registerArray[i][1];
                registerArray[i][1] = registerArray[j][1];
                registerArray[j][1] = tmp;
                String tmp1 = registerArray[i][3];
                registerArray[i][3] = registerArray[j][3];
                registerArray[j][3] = tmp1;
            }
        }
    }
}

```

```

        String tmp2 = registerArray[i][4];
        registerArray[i][4] = registerArray[j][4];
        registerArray[j][4] = tmp2;
        String tmp3 = registerArray[i][8];
        registerArray[i][8] = registerArray[j][8];
        registerArray[j][8] = tmp3;

    }
} else {
    break;
}

}

}

//registerArray의 1 index = 날짜, 3 index = 시작시간, 4 index = 퇴근시간, 8 index = 퇴근날짜 확인용
(디버그용 지우지 마세요!)
for (int i = 0; i < count; i++) {
    System.out.println("registerArray = (" + registerArray[i][1] + " , " + registerArray[i][3] + " , " +
registerArray[i][4] + " , " + registerArray[i][8] + ")");
}

//count = registerArray에서 위의 모든 조건에 해당한 데이터 row의 갯수(즉 원하는 날짜사이의 필요한
데이터들)
//workinghours에 registerArray의 버블 정렬한것의 시간데이터만 순서대로 넣어주고
hoursSubtractions에 to와 from의 시간차를 subHours()로 구해서 넣어준다.
for (int i = 0; i < count; i++) {
    workingHours.add(new String[]{registerArray[i][1], registerArray[i][3], registerArray[i][4]});
    System.out.println(workingHours.get(i)[0]);
    String from = workingHours.get(i)[1];
    String to = workingHours.get(i)[2];
    hoursSubtractions.add(new String[]{registerArray[i][1],
String.valueOf(subHours(registerArray[i][1], registerArray[i][8], from, to) - 1 - mealtime)}));
}

//위에서 구한 hoursSubtractions 이차원 배열을 Locat에 표시해준다.
for (int i = 0; i < count; i++) {
    System.out.println("hoursSubtractionsArray = (" + hoursSubtractions.get(i)[0] + " , " +
hoursSubtractions.get(i)[1]);
}

//기본급으로 계산한시간 (연장/야간 적용 없이) //휴일수당, 주휴수당 포함
//그 날 하루급여를 기준으로 계산
// 1 기본급
for (int i = 0; i < count; i++) {
    totalWorkingHours = Double.parseDouble(hoursSubtractions.get(i)[1]);
    System.out.println("하루 일한시간 : " + i + " = " + totalWorkingHours);
    oneDayWage[i] = totalWorkingHours * Double.parseDouble(totalArray[0][2]);

    //연장근무수당 하루 통상근무 시간인 8시간을 초과했을 때 또는 주급으로 따진다면 주 40시간 초과
    했을 때 적용(1.5배)
    if (totalWorkingHours > 8) {
        double extendedHours = totalWorkingHours - 8;
        double extendedWage = extendedHours * (Double.parseDouble(totalArray[0][2]) * 0.5);
        oneDayWage[i] = oneDayWage[i] + extendedWage;
    }
}

```

```

}

//야간 근무수당 밤 10시부터 오전 6시까지 일을 했을경우 그 사이의 일한 시간에 대하여 1.5를
받는다. 연장수당에 가산하여 받는다.(연장+야간 겹치기 가능)
Date tmpFromHours = timeDateFormat.parse(registerArray[i][1] + " 22:00"); //근무당일 밤 10시
Date today = format1.parse(registerArray[i][1]); //오늘 날짜
// 내일 날짜 구하기
assert today != null;
long tod = today.getTime() + 86400000; //근무당일 날짜 + 86400000(밀리초로 구한하루)
Date tomorrow = new Date(tod); //근무일다음날 날짜
String tom = format1.format(tomorrow); // 근무일다음날 0시 0분 0초를 format1을 String인
yyyy-MM-dd HH:mm:ss 날짜형식으로 만들어 tom(tomorrow의 약자)에 넣어줌
Date tmpToHours = timeDateFormat.parse(tom + " 06:00"); // tmpTohours = 근무일다음날 6시
Date tmpCurrentFromHours = timeDateFormat.parse(registerArray[i][1] + " 06:00"); // 근무당일
6시
Date fromHours = getFromHours(registerArray[i][1], registerArray[i][3]); // registerwork의
근무시작시간
Date ToHours = getToHours(registerArray[i][8], registerArray[i][4]); // registerwork의 근무종료시간
assert fromHours != null;
assert tmpToHours != null;
assert tmpFromHours != null;
assert ToHours != null;
assert tmpCurrentFromHours != null;
//야간수당 조건을 위한 시간의 기준들의 절대적 값을 Logcat 으로 확인(디버그용 지우지 마세요)
System.out.println(fromHours.getTime() + ", " + tmpFromHours.getTime() + ", " +
ToHours.getTime() + ", " + tmpToHours.getTime());
System.out.println(fromHours.getTime() + ", " + tmpCurrentFromHours.getTime() + ", " +
ToHours.getTime() + ", " + tmpCurrentFromHours.getTime());
System.out.println(fromHours.getTime() + ", " + tmpCurrentFromHours.getTime() + ", " +
ToHours.getTime() + ", " + tmpCurrentFromHours.getTime());

//야간 근무 수당 적용을 위한 조건들
//조건 1 : 근무시작시간 >= 근무당일 22시 && 근무시작시간 <= 근무일다음날 6시&& 근무종료시간
<= 근무일다음날 6시 && 근무종료시간 >= 근무당일 22시
//조건 2 : 근무시작시간 >= 근무당일 6시 && 근무시작시간 <= 근무당일 22시 && 근무종료시간 <=
근무일다음날 6시 && 근무종료시간 >= 근무당일 22시
//조건 3 : 근무시작시간 <= 근무당일 6시 && 근무종료시간 <= 근무당일 6시 && 근무종료시간 >
근무시작시간
//조건 4 : 근무시작시간 <= 근무당일 6시 && 근무종료시간 >= 근무당일 6시 && 근무종료시간 <=
근무당일 22시
if ((fromHours.getTime() >= tmpFromHours.getTime() && fromHours.getTime() <=
tmpToHours.getTime() && ToHours.getTime() <= tmpToHours.getTime() && ToHours.getTime() >=
tmpFromHours.getTime())) {
    double nightWorkingWage = totalWorkingHours * (Double.parseDouble(totalArray[0][2]) * 0.5);
    System.out.println("hours" + totalWorkingHours);
    oneDayWage[i] = oneDayWage[i] + nightWorkingWage;
} else if (fromHours.getTime() >= tmpCurrentFromHours.getTime() && fromHours.getTime() <=
tmpFromHours.getTime() && ToHours.getTime() <= tmpToHours.getTime() && ToHours.getTime() >=
tmpFromHours.getTime()) {
    double hours = Double.parseDouble(String.valueOf(ToHours.getTime() -
tmpFromHours.getTime())) / (1000 * 60 * 60);
    System.out.println("hours1 : " + i + " = " + hours);
    double nightWorkingWage = hours * (Double.parseDouble(totalArray[0][2]) * 0.5);
    oneDayWage[i] = oneDayWage[i] + nightWorkingWage;
} else if (fromHours.getTime() <= tmpCurrentFromHours.getTime() && ToHours.getTime() <=
tmpCurrentFromHours.getTime() && ToHours.getTime() > fromHours.getTime()) {
    double hours = Double.parseDouble(String.valueOf(ToHours.getTime() -
fromHours.getTime())) / (1000 * 60 * 60);
    double nightWorkingWage = hours * (Double.parseDouble(totalArray[0][2]) * 0.5);
}

```

```

        System.out.println("hours2" + hours);
        oneDayWage[i] = oneDayWage[i] + nightWorkingWage;
    } else if (fromHours.getTime() <= tmpCurrentFromHours.getTime() && ToHours.getTime() >=
tmpCurrentFromHours.getTime() && ToHours.getTime() <= tmpFromHours.getTime()) {
        double hours = Double.parseDouble(String.valueOf((tmpCurrentFromHours.getTime() -
fromHours.getTime())) / (1000 * 60 * 60));
        double nightWorkingWage = hours * (Double.parseDouble(totalArray[0][2]) * 0.5);
        System.out.println("hours3" + hours);
        oneDayWage[i] = oneDayWage[i] + nightWorkingWage;
    }

    //주말수당 검사 및 주말수당 적용
    //배열날짜(hoursSubtractions.get(i)[0]) == sat or sun 이라면 (그날의 날짜 == 토요일 또는
    일요일이라면)
    if (getDateDay(hoursSubtractions.get(i)[0], "yyyy-MM-dd", Locale.KOREA).equals("sat") ||
        getDateDay(hoursSubtractions.get(i)[0], "yyyy-MM-dd", Locale.KOREA).equals("sun")) {
        double tmpBreakWage = totalWorkingHours * (Double.parseDouble(totalArray[0][2]) * 0.5);
        oneDayWage[i] = oneDayWage[i] + tmpBreakWage;
    }

    System.out.println("하루 기본급 :" + i + " = " + oneDayWage[i]); //기본시급과 연장,주휴,야간을
    포함한 하루 기본급 Locat에 표시 디버그용 지우지마세요
}

//주휴 수당
double[] weekWorkingHours = new double[5];
try {
    for (int i = 0; i < count; i++) {
        //저장된 날짜가 그달의 몇주째인지 계산해서 weekNum(week number)에 넣어준다.
        int weekNum = getWeekOfYear(hoursSubtractions.get(i)[0]);
        System.out.println("weekNum = " + weekNum + ", " + hoursSubtractions.get(i)[0]); //디버그용
        지우지마세요
        String DateDay = getDateDay(hoursSubtractions.get(i)[0], "yyyy-MM-dd", Locale.KOREA);
        //근무당일날짜의 요일 getDateDay가 날짜를 그 날짜의 요일로 바꿔줌
        //받은 날짜가 1주차 날짜라면
        if (weekNum == 1) {
            String[] dayOfWeeks = totalArray[0][3].split(","); // totalArray에 기본적으로 근무를하기로
            정해진 요일들을 dayOfWeeks에 배열로 나누어 넣음
            System.out.println("dayOfWeeks1" + dayOfWeeks.length); //디버그용 지우지마세요
            for (int j = 0; j < 7; j++) {
                if (dayOfWeeks[j].equals(DateDay)) { //근무하기로 정해진 요일 == 근무당일날짜의 요일
                    firstWeek += 1; // firstweek 첫째주의 근무나간 횟수를 의미하는 int변수 +1
                    weekWorkingHours[0] = weekWorkingHours[0] +
                    Double.parseDouble(hoursSubtractions.get(i)[1]); // 그주의 일한시간 = 그주의 일한시간 +
                    근무당일일한시간
                    System.out.println("firstweek = " + weekWorkingHours[0]); //디버그용 지우지마세요
                }
            }
        }
        //받은 날짜가 2주차 날짜라면 (과정은 위와 동일)
        else if (weekNum == 2) {
            String[] dayOfWeeks = totalArray[0][3].split(",");
            System.out.println("dayOfWeeks2" + dayOfWeeks.length); //디버그용 지우지마세요
            for (int j = 0; j < 7; j++) {
                if (dayOfWeeks[j].equals(DateDay)) {
                    secondWeek += 1;
                    weekWorkingHours[1] = weekWorkingHours[1] +

```

```

Double.parseDouble(hoursSubtractions.get(i)[1]);
    System.out.println("secondweek = " + weekWorkingHours[1]);
}
}

} //받은 날짜가 3주차 날짜라면(과정은 위와 동일)
else if (weekNum == 3) {
    String[] dayOfWeeks = totalArray[0][3].split(",");
    System.out.println("dayOfWeeks3" + dayOfWeeks.length); //디버그용 지우지마세요
    for (int j = 0; j < 7; j++) {
        if (dayOfWeeks[j].equals(DateDay)) {
            thirdWeek += 1;
            weekWorkingHours[2] = weekWorkingHours[2] +
Double.parseDouble(hoursSubtractions.get(i)[1]);
                System.out.println("thirdweek = " + weekWorkingHours[2]); //디버그용 지우지마세요
            }
        }
    } //받은 날짜가 4주차 날짜라면(과정은 위와 동일)
else if (weekNum == 4) {
    String[] dayOfWeeks = totalArray[0][3].split(",");
    System.out.println("dayOfWeeks4" + dayOfWeeks.length); //디버그용 지우지마세요
    for (int j = 0; j < 7; j++) {
        if (dayOfWeeks[j].equals(DateDay)) {
            forthWeek += 1;
            weekWorkingHours[3] = weekWorkingHours[3] +
Double.parseDouble(hoursSubtractions.get(i)[1]);
                System.out.println("forthweek = " + weekWorkingHours[3]); //디버그용 지우지마세요
            }
        }
    } //받은 날짜가 5주차 날짜라면(과정은 위와 동일)
else if (weekNum == 5) {
    String[] dayOfWeeks = totalArray[0][3].split(",");
    System.out.println("dayOfWeeks5" + dayOfWeeks.length); //디버그용 지우지마세요
    for (int j = 0; j < 7; j++) {
        if (dayOfWeeks[j].equals(DateDay)) {
            fifthWeek += 1;
            weekWorkingHours[4] = weekWorkingHours[4] +
Double.parseDouble(hoursSubtractions.get(i)[1]);
                System.out.println("fifthweek = " + weekWorkingHours[4]); //디버그용 지우지마세요
            }
        }
    }
} //위의 try문이 실패할경우 즉 저장된 데이터가 없을경우
catch (Exception e) {
    e.printStackTrace();
}

//String[] dayOfWeeks = totalArray[0][3].split(",");
int dayofweekcount = 0;
double value;
for (String dayOfWeek : dayOfWeeks) {
    if (!dayOfWeek.equals("null")) {
        dayofweekcount += 1;
    }
}
//만약 그 grid의 개수가 totalarray의 요일의 개수와 일치하고 그 주의 총 근무시간이 15시간 이상이라면 주휴수당을 받아야한다.

```

```

// (1주일 총근로시간/40) * totalArray를 통해서 구한 근무시간 * 시급
if (firstWeek == dayofweekcount && weekWorkingHours[0] >= 15) {
    if (weekWorkingHours[0] >= 40) {
        weekWorkingHours[0] = 40;
    }
    value = (weekWorkingHours[0] / 40) * 8 * Double.parseDouble(totalArray[0][2]);
    extraMoney[0] = value;
    System.out.println("extraMoney 0 " + extraMoney[0]);
}
if (secondWeek == dayofweekcount && weekWorkingHours[1] >= 15) {
    if (weekWorkingHours[1] >= 40) {
        weekWorkingHours[1] = 40;
    }
    value = (weekWorkingHours[1] / 40) * 8 * Integer.parseInt(totalArray[0][2]);
    extraMoney[1] = value;
    System.out.println("extraMoney 1 " + extraMoney[1]);
}
if (thirdWeek == dayofweekcount && weekWorkingHours[2] >= 15) {
    if (weekWorkingHours[2] >= 40) {
        weekWorkingHours[2] = 40;
    }
    value = (weekWorkingHours[2] / 40) * 8 * Integer.parseInt(totalArray[0][2]);
    extraMoney[2] = value;
    System.out.println("extraMoney 2 " + extraMoney[2]);
}
if (forthWeek == dayofweekcount && weekWorkingHours[3] >= 15) {
    if (weekWorkingHours[3] >= 40) {
        weekWorkingHours[3] = 40;
    }
    value = (weekWorkingHours[3] / 40) * 8 * Integer.parseInt(totalArray[0][2]);
    extraMoney[3] = value;
    System.out.println("extraMoney 3 " + extraMoney[3]);
}
if (fifthWeek == dayofweekcount && weekWorkingHours[4] >= 15) {
    if (weekWorkingHours[4] >= 40) {
        weekWorkingHours[4] = 40;
    }
    value = (weekWorkingHours[4] / 40) * 8 * Integer.parseInt(totalArray[0][2]);
    extraMoney[4] = value;
    System.out.println("extraMoney 4 " + extraMoney[4]);
}
double extramoney = 0;
try {
    for (int i = 0; i < 5; i++) {
        try {
            extramoney += extraMoney[i];
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        System.out.println("extramoney " + extramoney);
    }
} catch (Exception e) {
    System.out.println("주휴수당은 없습니다.");
}

//extramoney = 총 주휴수당 여기다 모든 oneDayWage를 더하면 그 기간동안의 법정 최소한의
급여가 된다.
for (int i = 0; i < count; i++) {
    totalWage += oneDayWage[i];
}

```

```

        System.out.println(i + ":" + oneDayWage[i]);
    }
    totalWage = totalWage + extramoney;

    showWage.setText(String.format(Locale.KOREA, "%d 원", (int) totalWage));
    PreferenceManager.setString(getApplicationContext(), "suggestedSavedWage",
String.format(Locale.KOREA, "%d 원", (int) totalWage));
    System.out.println(String.format(Locale.KOREA, "%d 원", (int) totalWage));

} catch (Exception ignored) {
    System.out.println("무시됨");
}
}

}

```

MyApplication.java

```

package com.alba.lionproject1;

import android.app.Application;
import android.content.res.Configuration;

import androidx.annotation.NonNull;

import java.util.ArrayList;

public class Myapplication extends Application {

    @Override
    public void onCreate() {
        super.onCreate();
    }

    @Override
    public void onConfigurationChanged(@NonNull Configuration newConfig) {
        super.onConfigurationChanged(newConfig);
    }

    private String nameOfAlba = "알바를 선택하세요";
    private String photo_atPath = "";
    private String photo_afterPath = "";
    private String mainDateFrom = "";
    private String mainDateTo = "";
    private String date_To = "";
    private String gridDates = "";

    public String getNameOfAlba() {
        return nameOfAlba;
    }

    public void setNameOfAlba(String nameOfAlba) {
        this.nameOfAlba = nameOfAlba;
    }
}

```

```

public String getPhoto_atPath() {
    return photoAtPath;
}

public void setPhoto_atPath(String phoroPath) {
    this.photoAtPath = phoroPath;
}

public String getPhoto_afterPath() {
    return photoAfterPath;
}

public void setPhoto_afterPath(String photoAfterPath) {
    this.photoAfterPath = photoAfterPath;
}

public String getDate_To() {
    return dateTo;
}

public void setDate_To(String dateTo) {
    this.dateTo = dateTo;
}

public String getGridDates() {
    return gridDates;
}

public void setGridDates(String gridDates) {
    this.gridDates = gridDates;
}

public String getMainDateFrom() {
    return mainDateFrom;
}

public String getMainDateTo() {
    return mainDateTo;
}

public void setMainDateFrom(String mainDateFrom) {
    this.mainDateFrom = mainDateFrom;
}

public void setMainDateTo(String mainDateTo) {
    this.mainDateTo = mainDateTo;
}
}

```

MyGridAdapter.java

```

package com.alba.lionproject1;

import android.app.Activity;

```

```

import android.content.Context;
import android.database.Cursor;
import android.graphics.Color;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.Button;

import android.widget.TextView;

import androidx.annotation.NonNull;
import androidx.annotation.Nullable;
import androidx.core.content.ContextCompat;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.Date;
import java.util.List;
import java.util.Locale;

public class MyGridAdapter extends ArrayAdapter {
    List<Date> dates;
    Calendar currentDate;
    List<Events> events;
    LayoutInflater inflater;
    SimpleDateFormat dateFormat = new SimpleDateFormat("dd", Locale.KOREA);
    int count = 0;

    public MyGridAdapter(@NonNull Context context, List<Date> dates, Calendar currentDate,
    List<Events> events) {
        super(context, R.layout.single_cell_layout);

        this.dates = dates;
        this.currentDate = currentDate;
        this.events = events;
        inflater = LayoutInflater.from(context);
    }

    @Override
    public View getView(int position, @Nullable View convertView, @NonNull ViewGroup parent) {

        Date currentTime = Calendar.getInstance().getTime();
        Date monthDate = dates.get(position);
        Calendar dateCalendar = Calendar.getInstance();
        dateCalendar.setTime(monthDate);
        int DayNo = dateCalendar.get(Calendar.DAY_OF_MONTH);
        int displayMonth = dateCalendar.get(Calendar.MONTH) + 1;
        int displayYear = dateCalendar.get(Calendar.YEAR);
        int currentMonth = currentDate.get(Calendar.MONTH) + 1;
        int currentYear = currentDate.get(Calendar.YEAR);
        int currentDay = Integer.parseInt(dateFormat.format(currentTime));
    }
}

```

```

View view = convertView;
if (view == null) {
    view = inflater.inflate(R.layout.single_cell_layout, parent, false);

}

if (displayMonth == currentMonth && displayYear == currentYear) {
    //view.setBackgroundColor(getContext().getResources().getColor(R.color.green));
    view.setBackground(getContext().getResources().getDrawable(R.drawable.calendarborder, null));

} else {
    view.setBackground(getContext().getResources().getDrawable(R.drawable.offcalendarborder,
null));
    count += 1;

}

String writtenMonth = PreferenceManager.getString(getContext(), "time");
String cMonth = writtenMonth.substring(5, 6);
String cYear = writtenMonth.substring(0, 4);
if (Integer.parseInt(cYear) == currentYear && Integer.parseInt(cMonth) == currentMonth && DayNo ==
currentDay && count == 0) {
    view.setBackground(getContext().getResources().getDrawable(R.drawable.chosencalendar, null));

}

count = 0;

TextView Day_Number = view.findViewById(R.id.Calendar_day);
TextView EventNumber = view.findViewById(R.id.events_id);

Button btn_forAlbaName = ((MainActivity)
MainActivity.context_main).findViewById(R.id.btn_ChooseParttime);
String albaname = btn_forAlbaName.getText().toString();

String[] daysOfWeek = new String[7];
String test;
String albaday = "";
Cursor cursor = getTotalCursor();

for (int i = 0; i < cursor.getCount(); i++) {
    cursor.moveToPosition(i);

    if (cursor.getString(1).equals(albaname)) {
        daysOfWeek = cursor.getString(3).split(", ");
        break;
    }
}

for (int i = 0; i < 7; i++) {
    System.out.println(daysOfWeek[i]);
}
String date = currentYear + "-" + ChangeDateString(currentMonth) + "-" + ChangeDateString(DayNo);
try {

```

```

test = getDateDay(date, "yyyy-MM-dd", Locale.KOREA);
for (int i = 0; i < 7; i++) {
    if (test.equals(daysOfWeek[i]) && (displayMonth == currentMonth && displayYear == currentYear)) {
        EventNumber.setTextColor(Color.BLACK);
        EventNumber.setText("일바날");
        albaday = "일바날";
    }
}
} catch (Exception e) {
    e.printStackTrace();
}

//registerWorkDBHelper
String tmpMonth = "null";
String tmpDay = "null";
String tmpYear = "null";
Cursor cursor1 = getregisterCursor(); // registerWork database
for (int i = 0; i < cursor1.getCount(); i++) {
    cursor1.moveToPosition(i);
    if (albaname.equals(cursor1.getString(2))) {
        tmpYear = cursor1.getString(1).substring(0, 3 + 1);
        tmpMonth = cursor1.getString(1).substring(5, 6 + 1);
        tmpDay = cursor1.getString(1).substring(8, 9 + 1);
    }
    if (tmpDay.equals(ChangeDateString(DayNo)) &&
tmpMonth.equals(ChangeDateString(displayMonth)) && tmpYear.equals(Integer.toString(displayYear))) {
        EventNumber.setTextColor(ContextCompat.getColorStateList(getContext(), R.color.peach));
        String saved = "저장됨\n" + albaday;
        EventNumber.setText(saved);
        System.out.println("임시입력달 = " + tmpMonth + "\n임시입력일 = " + tmpDay);
    }
}
}

Day_Number.setText(String.valueOf(DayNo));
Calendar eventCalendar = Calendar.getInstance();
for (int i = 0; i < events.size(); i++) {
    eventCalendar.setTime(ConvertStringToDate(events.get(i).getDATE()));
}

}

//지워도 되는지 판단중(지워도 잘 작동 될거같음)
if (EventNumber.getText().toString().equals("저장됨\n일바날")) {
    Myapplication myapplication = (Myapplication) ((Activity) getContext()).getApplication();
    myapplication.setGridDates(myapplication.getGridDates() + " , " + displayYear + "-" + displayMonth
+ "-" + DayNo);
    System.out.println("setgridDates = " + myapplication.getGridDates());
    try {
        PreferenceManager.setString(getContext(), "gridDates", myapplication.getGridDates());
    } catch (Exception e) {
        e.printStackTrace();
    }
}

return view;
}

```

```

private Date ConvertStringToDate(String eventDate) {
    SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd", Locale.KOREAN);
    Date date = null;
    try {
        date = format.parse(eventDate);
    } catch (ParseException e) {
        e.printStackTrace();
    }
    return date;
}

@Override
public int getCount() {
    return dates.size();
}

@Override
public int getPosition(@Nullable Object item) {
    return dates.indexOf(item);
}

@Nullable
@Override
public Object getItem(int position) {
    return dates.get(position);
}

private Cursor getTotalCursor() {
    totalDB db = totalDB.getInstance(getApplicationContext());
    return db.getReadableDatabase().query(
        DataBases.CreatDB._TABLENAME0, null, null, null, null, null, DataBases.CreatDB._ID);
}

public String getDateDay(String date, String dateType, Locale locale) throws Exception {

    String day = "";
    SimpleDateFormat dateFormat = new SimpleDateFormat(dateType, locale);
    Date nDate = dateFormat.parse(date);

    Calendar cal = Calendar.getInstance();
    assert nDate != null;
    cal.setTime(nDate);

    int dayNum = cal.get(Calendar.DAY_OF_WEEK);

    switch (dayNum) {
        case 1:
            day = "sun";
            break;
        case 2:

```

```

        day = "mon";
        break;
    case 3:
        day = "tue";
        break;
    case 4:
        day = "wed";
        break;
    case 5:
        day = "thu";
        break;
    case 6:
        day = "fri";
        break;
    case 7:
        day = "sat";
        break;
    }

    return day;
}

public String ChangeDateString(int dayno) {
    String Dayno = Integer.toString(dayno);
    if (dayno < 10) {
        return "0" + Dayno;
    } else {
        return Dayno;
    }
}

private Cursor getregisterCursor() {
    registerWorkDBHelper db = registerWorkDBHelper.getInstance(getContext());
    return db.getReadableDatabase().query(
        registerWorkContract.registerWorkEntry.TABLE_NAME, null, null, null, null, null,
registerWorkContract.registerWorkEntry._ID);
}

}

```

partTimeContent.java

```

package com.alba.lionproject1;

import androidx.appcompat.app.AppCompatActivity;

import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;

```

```

import android.os.Bundle;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.CheckBox;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;
import android.widget.Toast;
import static com.alba.lionproject1.registerWorkContract.registerWorkEntry.TABLE_NAME;
import java.util.Objects;

public class partTimeContent extends AppCompatActivity {
    String numberofpeople = "";
    public static Context mContext;
    String[] cbResult = new String[8];
    String parttimename = "";
    String hourlywage = "";
    String dayofweek = "";
    String startingwork = "";
    String finishingwork = "";
    int mealtime = 0;
    RadioGroup rb_group;
    int id;
    String[] totalDBArray = new String[6];
    String[] splitarray = new String[7];

    public static Context getmContext() {
        return mContext;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_part_time_content);
        mContext = this;
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
                WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_part_time_content);
        // 배열 초기화
        for (int i = 0; i < 8; i++) {
            cbResult[i] = "null";
        }

        //화면의 모든 정보 아이템들 캐스팅
        final TextView txt_parttimename = findViewById(R.id.txt_ParttimeName);
        final TextView txt_hourlywage = findViewById(R.id.txt_hourlypay);
        final TextView txt_ptcfrom = findViewById(R.id.txt_pFrom);
        final TextView txt_ptcTo = findViewById(R.id.txt_pTo);
        final CheckBox cb_mon = findViewById(R.id.cb_mon);
        final CheckBox cb_tues = findViewById(R.id.cb_tues);
        final CheckBox cb_wed = findViewById(R.id.cb_wed);
        final CheckBox cb_thu = findViewById(R.id.cb_thurs);
        final CheckBox cb_fri = findViewById(R.id.cb_fri);
        final CheckBox cb_sat = findViewById(R.id.cb_sat);
        final CheckBox cb_sun = findViewById(R.id.cb_sun);
        final RadioButton rb_yes = findViewById(R.id.rb_yes);
    }
}

```

```

final RadioButton rb_no = findViewById(R.id.rb_no);
final EditText txt_mealtime = findViewById(R.id.txt_mealtime);
final ImageButton helpworkplace = findViewById(R.id.btn_helpWorkplace);
final ImageButton helpbreaktime = findViewById(R.id.btn_helpBreaktime);

// 전화면으로 부터의 ID(데이터베이스 수정시 버튼 구분을 위한 번호) 받기
Intent intent = getIntent();
id = Objects.requireNonNull(intent.getExtras()).getInt("id");
id = Integer.parseInt(PreferenceManager.getString(getApplicationContext(), "spid"));
totalDBArray = intent.getExtras().getStringArray("totaldbarray");
//DB로 받아온 정보들을 화면에 표시(기준저장정보)
assert totalDBArray != null;
txt_parttimename.setText(totalDBArray[0]);
txt_hourlywage.setText(totalDBArray[1]);
txt_ptcfrom.setText(totalDBArray[3]);
txt_ptcTo.setText(totalDBArray[4]);
txt_mealtime.setText(totalDBArray[6]);

splitarray = totalDBArray[2].split(",");
for (int i = 0; i < 7; i++) {
    if (splitarray[i].equals("mon")) {
        cbResult[0] = "mon";
        cb_mon.setChecked(true);
    }
    if (splitarray[i].equals("tue")) {
        cbResult[1] = "tue";
        cb_tues.setChecked(true);
    }
    if (splitarray[i].equals("wed")) {
        cbResult[2] = "wed";
        cb_wed.setChecked(true);
    }
    if (splitarray[i].equals("thu")) {
        cbResult[3] = "thu";
        cb_thu.setChecked(true);
    }
    if (splitarray[i].equals("fri")) {
        cbResult[4] = "fri";
        cb_fri.setChecked(true);
    }
    if (splitarray[i].equals("sat")) {
        cbResult[5] = "sat";
        cb_sat.setChecked(true);
    }
    if (splitarray[i].equals("sun")) {
        cbResult[6] = "sun";
        cb_sun.setChecked(true);
    }
}

if (totalDBArray[5].equals("true")) {
    rb_yes.setChecked(true);
    numberofpeople = "true";
} else if (totalDBArray[5].equals("false")) {
    rb_no.setChecked(true);
    numberofpeople = "false";
}

```

```
//체크박스 클릭 및 체크박스로부터 요일 데이터 얻기
cb_mon.setOnClickListener(view -> {
    if (cb_mon.isChecked()) {
        cbResult[0] = "mon";
    } else {
        cbResult[0] = "null";
    }
});

cb_tues.setOnClickListener(view -> {
    if (cb_tues.isChecked()) {
        cbResult[1] = "tue";
    } else {
        cbResult[1] = "null";
    }
});

cb_wed.setOnClickListener(view -> {
    if (cb_wed.isChecked()) {
        cbResult[2] = "wed";
    } else {
        cbResult[2] = "null";
    }
});

cb_thu.setOnClickListener(view -> {
    if (cb_thu.isChecked()) {
        cbResult[3] = "thu";
    } else {
        cbResult[3] = "null";
    }
});

cb_fri.setOnClickListener(view -> {
    if (cb_fri.isChecked()) {
        cbResult[4] = "fri";
    } else {
        cbResult[4] = "null";
    }
});

cb_sat.setOnClickListener(view -> {
    if (cb_sat.isChecked()) {
        cbResult[5] = "sat";
    } else {
        cbResult[5] = "null";
    }
});

cb_sun.setOnClickListener(view -> {
    if (cb_sun.isChecked()) {
        cbResult[6] = "sun";
    } else {
        cbResult[6] = "null";
    }
});
```

```

helpworkplace.setOnClickListener(view -> {
    Intent help_workplace_intent = new Intent(partTimeContent.this,
partTimeContent_help_Workplace.class);
    startActivity(help_workplace_intent);
});

helpbreaktime.setOnClickListener(view -> {
    Intent help_breaktime_intent = new Intent(partTimeContent.this,
partTimeContent_help_Breaktime.class);
    startActivity(help_breaktime_intent);
});

//저장 버튼 캐스팅과 클릭시 이벤트
Button btn_Save = (Button) findViewById(R.id.btn_pSave);
btn_Save.setOnClickListener(view -> {
    //알바이름,시간당 임금
    dayofweek = ShowResultOfDaysOfWeek();
    parttimename = txt_parttimename.getText().toString();
    hourlywage = txt_hourlywage.getText().toString();
    //근무시작과 끝시간텍스트
    startingwork = txt_ptcfrom.getText().toString();
    finishingwork = txt_ptcTo.getText().toString();

    try {
        mealtime = Integer.parseInt(txt_mealtime.getText().toString());
    } catch (Exception ex) {
        mealtime = 0;
    }

    if (dayofweek.equals("null,null,null,null,null,null")) || parttimename.equals("null") ||
hourlywage.equals("null") || startingwork.equals("null") || finishingwork.equals("null") ||
numberofpeople.equals("null")) {
        Toast.makeText(getApplicationContext(), "잘봐봐 안쓸거 있어", Toast.LENGTH_LONG).show();
    } else if (txt_mealtime.getText().toString().isEmpty() || !isNumber(txt_mealtime.getText().toString()))
    {
        Toast.makeText(getApplicationContext(), "시간은 숫자로 써야되", Toast.LENGTH_LONG).show();
    } else {

        ContentValues myContentValues = new ContentValues();
        myContentValues.put(DataBases.CreatDB.PARTTIMENAME, parttimename);
        myContentValues.put(DataBases.CreatDB.HOURLYWAGE, hourlywage);
        myContentValues.put(DataBases.CreatDB.DAYOFWEEK, dayofweek);
        myContentValues.put(DataBases.CreatDB.WORKINGSTART, startingwork);
        myContentValues.put(DataBases.CreatDB.WORKINGFINISH, finishingwork);
        myContentValues.put(DataBases.CreatDB.NUMBER_OF_PEOPLE, numberofpeople);
        myContentValues.put(DataBases.CreatDB.MEAL_TIME, mealtime);

        SQLiteDatabase db = totalDB.getInstance(getApplicationContext()).getWritableDatabase();
        //long newRowId = db.insert(DataBases.CreatDB._TABLENAME0,null,myContentValues);
        long newRowId = db.update(DataBases.CreatDB._TABLENAME0, myContentValues,
DataBases.CreatDB._ID + " = " + id, null);
        if (newRowId == 0) {
            Toast.makeText(getApplicationContext(), "저장에 문제 발생", Toast.LENGTH_SHORT).show();
        } else {
            Toast.makeText(getApplicationContext(), "저장 성공", Toast.LENGTH_SHORT).show();
        }
    }
}

```

```

        }
        long rowid = 0;
        if(!totalDBArray[0].equals(parttimename))
        {
            SQLiteDatabase rwDB =
registerWorkDBHelper.getINSTANCE(getApplicationContext()).getWritableDatabase();
            ContentValues values = new ContentValues();
            values.put(registerWorkContract.registerWorkEntry.COLUMN_NAME_TITLE,parttimename);
            rowid =
rwDB.update(TABLE_NAME,values,registerWorkContract.registerWorkEntry.COLUMN_NAME_TITLE + " "
= "?",new String[]{totalDBArray[0]});
        }
        System.out.println("rowid = "+ rowid);

        Intent intent1 = new Intent(partTimeContent.this, SelectParttime.class);
        getmContext().startActivity(intent1);
        finish();
    }
});

//라디오버튼,라디오그룹 (5인이상사업장)에 대한 처리
rb_group = (RadioGroup) findViewById(R.id.rb_group);

rb_group.setOnCheckedChangeListener((radioGroup, i) -> {
    if (i == R.id.rb_yes) {
        numberofpeople = "true";
    } else if (i == R.id.rb_no) {
        numberofpeople = "false";
    }
});

//취소 버튼 캐스팅과 클릭시 이벤트
Button btn_Cancel = (Button) findViewById(R.id.btn_pCancel);
btn_Cancel.setOnClickListener(view -> {
    Intent intent12 = new Intent(partTimeContent.this, SelectParttime.class);
    getmContext().startActivity(intent12);
    finish();
});

//몇시부터 몇시까지 고정적으로 하는지 설정할수 있게 해주는 버튼 btn_tpFrom 그리고 클릭시 이벤트
Button btn_From = (Button) findViewById(R.id.btn_tpFrom);
btn_From.setOnClickListener(view -> {
    Intent intent13 = new Intent(partTimeContent.this, Timepicker.class);
    intent13.putExtra("view", "ptcfrom");
    startActivityForResult(intent13);
});

//몇시부터 몇시까지 일을 고정적으로 하는지 설정할수 있게 해주는 버튼 btn_tpTo 그리고 클릭시 이벤트
Button btn_To = (Button) findViewById(R.id.btn_tpTo);
btn_To.setOnClickListener(view -> {
    Intent intent14 = new Intent(partTimeContent.this, Timepicker.class);
    intent14.putExtra("view", "ptcto");
    startActivityForResult(intent14);
});

}

private String ShowResultOfDaysofWeek() {
    StringBuilder Results = new StringBuilder();
    for (int i = 0; i < 7; i++) {
        if (i == 0) {

```

```

        Results.append(cbResult[i]);
    } else {
        Results.append(".").append(cbResult[i]);
    }
}
return Results.toString();
}

private boolean isNumber(String toCheck)
{
    char tmp;
    boolean output = true;
    for(int i = 0;i<toCheck.length();i++)
    {
        tmp = toCheck.charAt(i);
        if(!Character.isDigit(tmp))
        {
            output = false;
        }
    }
    return output;
}

}

```

parttimeContent_help_Breaktime.java

```

package com.alba.lionproject1;

import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class partTimeContent_help_Breaktime extends AppCompatActivity {
    Button helpOk;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.part_time_content_help_breaktime);

        String help_Breaktime = "1. 근로시간이 4시간 이상인 경우 30분이상, 8시간 이상인 경우 1시간 이상의  
휴게시간을 제공해야합니다.\n\n" +  
        "2. 휴게시간은 근로가 이루어지는 시간이 아니기 때문에, 별도 사업장 규정이 없다면 무급이  
원칙입니다.\n\n" +  
        "3. 휴게시간은 근로자가 자유롭게 이용할 수 있습니다.\n\n" +  
        "※ 해당 근로기준법은 근로기준법 제 54조 <휴게>에 관한 사항으로 5인 미만 사업장과 5인 이상의  
사업장 모두 적용됩니다.";
    }
}

```

```

        TextView HelpBreaktime = findViewById(R.id.txt_help_breaktime);
        HelpBreaktime.setText(help_Breaktime);

        helpOk = findViewById(R.id.btn_help_breaktime_ok);
        helpOk.setOnClickListener(view -> finish());
    }
}

```

partTimeContent_help_Workplace.java

```

package com.alba.lionproject1;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class partTimeContent_help_Workplace extends AppCompatActivity {
    Button helpOk;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.part_time_content_help_workplace);

        String help_Workplace =
            "위의 표에서 해당하는 부분은 근로기준법에 의해 근로자가 보호를 받을 수 있습니다.\n\n" +
            "1. 5인 이상 사업장은 상시근로자 수가 5인 이상인 사업장이고 5인 미만 사업장은 상시근로자
        수가 5인 미만인 사업장입니다.\n\n" +
            "2. 상시 근로자 수 계산방법은 1개월 동안 동원된 총 근로자 ÷ 1개월 동안 사업일 수
        입니다.\n\n" +
            "※ 1일 근로자 수가 5명 이상인 날이 50%를 넘으면 계산상 상시근로자가 5인 미만이더라도
        5인 이상 사업장으로 간주됩니다.";

        TextView HelpWorkPlace = findViewById(R.id.txt_help_workplace);
        HelpWorkPlace.setText(help_Workplace);

        helpOk = findViewById(R.id.btn_help_workplace_ok);
        helpOk.setOnClickListener(view -> finish());
    }

    helpOk = findViewById(R.id.btn_help_breaktime_ok);
    helpOk.setOnClickListener(view -> finish());
}
}

```

PreferenceManager.java

```
package com.alba.lionproject1;

import android.content.Context;
import android.content.SharedPreferences;

// 데이터 저장 및 로드 클래스

public class PreferenceManager {

    public static final String PREFERENCES_NAME = "rebuild_preference";

    private static final String DEFAULT_VALUE_STRING = "";

    private static SharedPreferences getPreferences(Context context) {

        return context.getSharedPreferences(PREFERENCES_NAME, Context.MODE_PRIVATE);
    }

    public static void setString(Context context, String key, String value) {

        SharedPreferences prefs = getPreferences(context);

        SharedPreferences.Editor editor = prefs.edit();

        editor.putString(key, value);

        editor.apply();
    }

    public static void setInt(Context context, String key, int value) {

        SharedPreferences prefs = getPreferences(context);

        SharedPreferences.Editor editor = prefs.edit();

        editor.putInt(key, value);

        editor.apply();
    }

    public static String getString(Context context, String key) {

        SharedPreferences prefs = getPreferences(context);

        return prefs.getString(key, DEFAULT_VALUE_STRING);
    }
}
```

ProbationPeriod_help.java

```

package com.alba.lionproject1;

import android.os.Bundle;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class ProbationPeriod_help extends AppCompatActivity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.probation_period_help);

        String ProbationPeriod_mean = "수습기간이란 법적으로 근로계약을 할 때 1년 이상의 근무를 약속하여
" +
                "일을 시작한 뒤로부터 3개월 이내가 법적수습기간으로 볼 수 있습니다.';

        String ProbationPeriod_calculate = "최저 급액에서 10%를 차감한 금액입니다. " +
                "예) 시급으로 8000원을 받는다면 시급 8000원에서 10%인 800원을 뺀 7200원을 시급으로
받게됩니다.\n\n" +
                "단, 단순 노무 업무로 고용노동부장관이 정하여 고시한 직종에 종사하는 근로자의 경우" +
                " 1년 이상 근무 계약을 하더라도 무조건 수습기간이 적용되지는 않습니다.\n\n" +
                "그러므로 단순 노무 종사자의 경우 수습기간의 상관없이 최저임금을 100%를 받을 수 있습니다.';

        TextView Probation_Period_Mean = findViewById(R.id.txt_probation_period_mean);
        Probation_Period_Mean.setText(ProbationPeriod_mean);

        TextView Probation_Period_Calculate = findViewById(R.id.txt_probation_period_calculate);
        Probation_Period_Calculate.setText(ProbationPeriod_calculate);

        Button Probation_Period = findViewById(R.id.btn_help_probationperiod_ok);
        Probation_Period.setOnClickListener(view -> finish());
    }
}

```

registerWork.java

```

package com.alba.lionproject1;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.FileProvider;
import android.Manifest;
import android.app.Activity;
import android.app.AlertDialog;
import android.content.ActivityNotFoundException;
import android.content.ContentValues;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.database.Cursor;

```

```

import android.database.sqlite.SQLiteDatabase;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Matrix;
import android.media.ExifInterface;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.provider.MediaStore;
import android.provider.Settings;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import android.widget.Toast;
import com.google.android.material.textfield.TextInputEditText;
import java.io.File;
import java.io.IOException;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;
import java.util.Objects;

import static com.alba.lionproject1.registerWorkContract.registerWorkEntry.TABLE_NAME;

public class registerWork extends AppCompatActivity implements View.OnClickListener {
    final String TAG = getClass().getSimpleName();
    int id;
    String NEXTDATE;
    Cursor cursor;
    ImageView imageView_at;
    ImageView imageView_after;
    Button cameraBtn_at;
    Button cameraBtn_after;
    String mCurrentPhotoPath;
    String calendardate, month, year;
    public static Context context;
    static final int REQUEST_TAKE_PHOTO = 1;
    int dividePath = 0;
    String[] registerArray = new String[9];
    String[][] totalArray = new String[20][8];
    String date = "";
    String title = "";
    String from = "";
    String to = "";
    String memo = "";
    File path_at = null;
    File path_after = null;
    double timeSubtraction = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register_work);
        context = this;
    }
}

```

```

//전역변수
final Myapplication myapplication = (Myapplication) getApplication();

for(int i=0;i<20;i++)
{
    for(int j=0;j<8;j++) {
        totalArray[i][j] = "";
    }
}
for (int i = 0; i < 9; i++) {
    registerArray[i] = "";
}

Intent intent = getIntent();
calendardate = Objects.requireNonNull(intent.getExtras()).getString("date");
month = intent.getExtras().getString("month");
year = intent.getExtras().getString("year");
id = Objects.requireNonNull(intent.getExtras()).getInt("id");

// 날씨, 이름, 시간, 메모 캐스팅
final TextView txt_date = (TextView) findViewById(R.id.txt_date);
final TextView albaname = (TextView) findViewById(R.id.albaName);
final TextView mFromEditText = (TextView) findViewById(R.id.txt_rwFrom);
final TextView mToEditText = (TextView) findViewById(R.id.txt_rwTo);
final TextInputEditText mMemoEditText = findViewById(R.id.txt_memo);
final TextView txt_rwDateFrom = (TextView) findViewById(R.id.txt_rwDateFrom);
final Button btn_rwDateTo = (Button) findViewById(R.id.btn_rwDateTo);
final TextView txt_rwDateTo = (TextView) findViewById(R.id.txt_rwDateTo);

imageView_at = findViewById(R.id.iv_picture_at_work);
imageView_after = findViewById(R.id.iv_picture_out_work);
cameraBtn_at = findViewById(R.id.btn_takepicture);
cameraBtn_after = findViewById(R.id.btn_takepicture_afterWork);

txt_date.setText(calendardate); // 날짜 텍스트 창에 날짜 넣기
txt_rwDateFrom.setText(calendardate);
Button albabutton = ((MainActivity)
MainActivity.context_main).findViewById(R.id.btn_ChooseParttime);
albaname.setText(albabutton.getText()); // 알바 이름 텍스트 창에 메인에 있는 알바1 버튼 이름
가져와서 넣기

//rwDate , rwFrom initialize
int count = 0;
Cursor totalDBCursor = getTotalCursor();
for (int i = 0; i < totalDBCursor.getCount(); i++) // n^2
{
    if (i > 0 && !(totalDBCursor.getString(1).equals(albabutton.getText().toString())))
    {
        i = i - 1;
    }
    totalDBCursor.moveToPosition(count);
    if (count < totalDBCursor.getCount())
        count = count + 1;
} else {
    break;
}

```

```

        }
        System.out.println(totalDBCursor.getString(1) + " , " + albabutton.getText().toString());
        if (totalDBCursor.getString(1).equals(albabutton.getText().toString())) {
            for (int j = 0; j < 8; j++) {
                totalArray[i][j] = totalDBCursor.getString(j);
                System.out.println("total : " + totalArray[i][j]);
            }
        }
    }

mFromEditText.setText(totalArray[0][4]);
mToEditText.setText(totalArray[0][5]);

//calendardate에서 하루 추가한 값 받기
SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd", Locale.KOREA);

try {
    Date toDate = format.parse(calendardate);
    assert toDate != null;
    long date = toDate.getTime() + 86400000;
    NEXTDATE = format.format(date);

} catch (ParseException e) {
    e.printStackTrace();
}

// DB에 저장된 데이터를 배열에 저장하는 코드
count = 0;
cursor = getregisterCursor();
for (int i = 0; i < cursor.getCount(); i++) {
    // row
    cursor.moveToPosition(i);
    // db에서 원하는 정보 가져오기
    if (txt_date.getText().toString().equals(cursor.getString(1)) &&
albaname.getText().toString().equals(cursor.getString(2))) {
        for (int j = 0; j < 9; j++) {
            // column
            registerArray[j] = cursor.getString(j);
        }
        count += 1;
        break;
    }
}

for (int i = 0; i < 9; i++) {
    System.out.println(registerArray[i]);
}

// DB 데이터가 저장된 배열로 화면에 텍스트 표시
if (count != 0) {
    assert registerArray != null;
    mFromEditText.setText(registerArray[3]);
    mToEditText.setText(registerArray[4]);
    mMemoEditText.setText(registerArray[5]);
    txt_rwDateTo.setText(registerArray[8]);
    if (!registerArray[6].equals("")) {

```

```

try {
    // 비트맵 이미지로 가져온다
    String atimgPath = registerArray[6].substring(7);
    Bitmap atlImage = BitmapFactory.decodeFile(atimgPath);

    // 이미지를 상황에 맞게 회전시킨다(at)
    ExifInterface exif_at = new ExifInterface(atimgPath);
    int exifOrientation_at = exif_at.getAttributeInt(ExifInterface.TAG_ORIENTATION,
ExifInterface.ORIENTATION_NORMAL);
    int exifDegree_at = exifOrientationToDegrees(exifOrientation_at);
    atlImage = rotate(atlImage, exifDegree_at);

    // 변환된 이미지 사용
    imageView_at.setImageBitmap(atlImage);
} catch (IOException ex) {
    Toast.makeText(this, "사진 변환에 오류가 발생하였습니다: " + ex.getLocalizedMessage(),
Toast.LENGTH_LONG).show();
}

if (!registerArray[7].equals("")) {
    try {
        // 비트맵 이미지로 가져온다(after)
        String afterimgPath = registerArray[7].substring(7);
        Bitmap afterImage = BitmapFactory.decodeFile(afterimgPath);
        // 이미지를 상황에 맞게 회전시킨다(after)
        ExifInterface exif_after = new ExifInterface(afterimgPath);
        int exifOrientation_after = exif_after.getAttributeInt(ExifInterface.TAG_ORIENTATION,
ExifInterface.ORIENTATION_NORMAL);
        int exifDegree_after = exifOrientationToDegrees(exifOrientation_after);
        afterImage = rotate(afterImage, exifDegree_after);
        // 변환된 이미지 사용(after)
        imageView_after.setImageBitmap(afterImage);
    } catch (IOException ex) {
        Toast.makeText(this, "사진 변환에 오류가 발생하였습니다: " + ex.getLocalizedMessage(),
Toast.LENGTH_LONG).show();
    }
}

//몇시 부터 몇시까지 일을 했는지 설정할 수 있게 해주는 버튼 btn_rwFrom 캐스팅과 클릭 시 이벤트
Button btn_From = (Button) findViewById(R.id.btn_rwFrom);
btn_From.setOnClickListener(view -> {
    Intent intent12 = new Intent(registerWork.this, Timepicker.class);
    intent12.putExtra("view", "rwfrom");
    startActivityForResult(intent12, 1);
});

//몇시 부터 몇시까지 일을 했는지 설정할 수 있게 해주는 버튼 btn_btn_rwTo 캐스팅과 클릭 시 이벤트
Button btn_To = (Button) findViewById(R.id.btn_rwTo);
btn_To.setOnClickListener(view -> {
    Intent intent1 = new Intent(registerWork.this, Timepicker.class);
    intent1.putExtra("view", "rwto");
    startActivityForResult(intent1, 1);
});

//저장 버튼 btn_rwSave 캐스팅과 클릭 시 이벤트

```

```

Button btn_Save = (Button) findViewById(R.id.btn_rwSave);
btn_Save.setOnClickListener((View.OnClickListener) view -> {

    timeSubtraction = TimeSubtraction(mFromEditText.getText().toString(),
    mEditText.getText().toString());

    if ((txt_rwDateFrom.getText().toString().equals(txt_rwDateTo.getText().toString()) &&
    timeSubtraction > 0)
        || !txt_rwDateFrom.getText().toString().equals(txt_rwDateTo.getText().toString()) &&
    timeSubtraction < 0) {
        // 날짜 텍스트, 알바이름
        date = txt_date.getText().toString();
        title = albaname.getText().toString();
        // 근무시작시간과 끝시간 텍스트
        from = mFromEditText.getText().toString();
        to = mEditText.getText().toString();
        // 메모창 텍스트
        memo = Objects.requireNonNull(mMemoEditText.getText()).toString();

        if (date.equals("") || title.equals("") || from.equals("") || to.equals("") || (path_at == null &&
registerArray[6].equals("")) && (path_after == null && registerArray[7].equals("")))) {
            Toast.makeText(getApplicationContext(), "항목들을 다 채웠는지 확인해봐",
Toast.LENGTH_LONG).show();
        } else {
            System.out.println();
            //이전 mainActivity 종료
            Activity mainActivity = (MainActivity) MainActivity.main_activity;
            mainActivity.finish();

            ContentValues contentValues = new ContentValues();
            contentValues.put(registerWorkContract.registerWorkEntry.COLUMN_NAME_DATE, date);
            contentValues.put(registerWorkContract.registerWorkEntry.COLUMN_NAME_TITLE, title);
            contentValues.put(registerWorkContract.registerWorkEntry.COLUMN_NAME_FROM, from);
            contentValues.put(registerWorkContract.registerWorkEntry.COLUMN_NAME_TO, to);
            contentValues.put(registerWorkContract.registerWorkEntry.COLUMN_NAME_MEMO, memo);

            try {
                myapplication.setPhotoAtPath(Uri.fromFile(path_at).toString());
            } catch (Exception ex) {
                myapplication.setPhotoAtPath(registerArray[6]);
            }
            try {
                myapplication.setPhotoAfterPath(Uri.fromFile(path_after).toString());
            } catch (Exception ex) {
                myapplication.setPhotoAfterPath(registerArray[7]);
            }

            contentValues.put(registerWorkContract.registerWorkEntry.COLUMN_NAME_ATPHOTOPATH,
myapplication.getPhotoAtPath());

            contentValues.put(registerWorkContract.registerWorkEntry.COLUMN_NAME_AFTERRPHOTOPATH,
myapplication.getPhotoAfterPath());
            contentValues.put(registerWorkContract.registerWorkEntry.COLUMN_NAME_TODATE,
txt_rwDateTo.getText().toString());
            myapplication.setPhotoAfterPath("");
            myapplication.setPhotoAtPath("");

            SQLiteDatabase db =

```

```

registerWorkDBHelper.getInstance(getApplicationContext()).getWritableDatabase();

        if (registerArray[1].equals(date)) {
            db.update(TABLE_NAME, contentValues, registerWorkContract.registerWorkEntry._ID + " = "
" + registerArray[0], null);
        } else {
            db.insert(TABLE_NAME, "null", contentValues);
        }

        SaveEvent(calendardate, month, year);
        Intent intent13 = new Intent(registerWork.this, MainActivity.class);
        db.close();
        startActivity(intent13);
        finish();
    }
} else {
    Toast.makeText(getApplicationContext(), "근무시작 시간 그리고 근무종료 시간과 날짜를
확인해봐", Toast.LENGTH_LONG).show();
}
});

//취소 버튼 btn_rwCancel 캐스팅과 클릭시 이벤트
Button btn_Cancel = (Button) findViewById(R.id.btn_rwCancel);
btn_Cancel.setOnClickListener(view -> finish());

//-----카메라 찰칵버튼 관련-----
// 카메라 버튼에 리스터 추가
cameraBtn_at.setOnClickListener((View.OnClickListener) this);
cameraBtn_after.setOnClickListener((View.OnClickListener) this);

// 6.0 마쉬멜로우 이상일 경우에는 권한 체크 후 권한 요청
if (checkSelfPermission(Manifest.permission.CAMERA) ==
PackageManager.PERMISSION_GRANTED &&
checkSelfPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE) ==
PackageManager.PERMISSION_GRANTED) {
    Log.d(TAG, "권한 설정 완료");
} else {
    Log.d(TAG, "권한 설정 요청");
    ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.CAMERA,
Manifest.permission.WRITE_EXTERNAL_STORAGE}, 1);
}

btn_rwDateTo.setOnClickListener(view -> {
    Intent intent14 = new Intent(registerWork.this, getDatePopUp.class);
    intent14.putExtra("view", "rwdateTo");
    intent14.putExtra("nextdate", NEXTDATE);
    intent14.putExtra("currentdate", calendardate);
    startActivity(intent14);
});

private double TimeSubtraction(String from, String to) {
    try {
        String[] organizedHoursFrom = from.split(" ");
        String[] organizedHoursTo = to.split(" ");

```

```

        if (organizedHoursFrom[0].equals("오후")) {
            organizedHoursFrom[1] = Integer.toString(Integer.parseInt(organizedHoursFrom[1]) + 12);
        }

        if (organizedHoursTo[0].equals("오후")) {
            organizedHoursTo[1] = Integer.toString(Integer.parseInt(organizedHoursTo[1]) + 12);
        }

        if (Integer.parseInt(organizedHoursFrom[1]) < 10) {
            organizedHoursFrom[1] = "0" + organizedHoursFrom[1];
        }

        if (Integer.parseInt(organizedHoursTo[1]) < 10) {
            organizedHoursTo[1] = "0" + organizedHoursTo[1];
        }

        from = organizedHoursFrom[1] + ":" + organizedHoursFrom[3];
        to = organizedHoursTo[1] + ":" + organizedHoursTo[3];

        SimpleDateFormat format = new SimpleDateFormat("HH:mm", Locale.KOREA);

        Date fromDate = format.parse(from);
        Date toDate = format.parse(to);

        assert fromDate != null;
        assert toDate != null;

        return (double) (toDate.getTime() - fromDate.getTime());
    } catch (Exception ex) {
        ex.printStackTrace();
        return 0;
    }
}

// 권한 요청
@Override
public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull
int[] grantResults) {
    super.onRequestPermissionsResult(requestCode, permissions, grantResults);
    Log.d(TAG, "onRequestPermissionsResult");
    if (grantResults[0] == PackageManager.PERMISSION_GRANTED && grantResults[1] ==
PackageManager.PERMISSION_GRANTED) {
        Log.d(TAG, "Permission: " + permissions[0] + "was " + grantResults[0]);
    } else{
        if (checkSelfPermission(Manifest.permission.CAMERA) !=
PackageManager.PERMISSION_GRANTED ||
checkSelfPermission(Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED) {
            AlertDialog.Builder localBuilder = new AlertDialog.Builder(this);

            localBuilder.setTitle("권한 설정")
                .setMessage("권한 거절로 인해 일부기능이 제한됩니다.")
                .setPositiveButton("권한 설정하려 가기", new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface paramAnonymousDialogInterface, int
paramAnonymousInt) {
                        try {

```

```

        Intent intent = new Intent(Settings.ACTION_APPLICATION_DETAILS_SETTINGS)
            .setData(Uri.parse("package:" + getPackageName()));
        startActivity(intent);
    } catch (ActivityNotFoundException e) {
        e.printStackTrace();
        Intent intent = new
Intent(Settings.ACTION_MANAGE_APPLICATIONS_SETTINGS);
        startActivity(intent);
    }
}
.setNegativeButton("취소하기", (paramAnonymousDialogInterface, paramAnonymousInt) ->
{
})
.create()
.show();
}
}

// 버튼 onClick리스너 처리 부분
@Override
public void onClick(View view) {
    if (view.getId() == R.id.btn_takepicture) {
        dividePath = 1;
        dispatchTakePictureIntent();

    }
    if (view.getId() == R.id.btn_takepicture_afterWork) {
        dividePath = 2;
        dispatchTakePictureIntent();

    }
}

private void dispatchTakePictureIntent() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    // Ensure that there's a camera activity to handle the intent
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        // Create the File where the photo should go
        File photoFile = null;
        try {
            photoFile = createImageFile();
            if (dividePath == 1) {
                path_at = photoFile;
            } else if (dividePath == 2) {
                path_after = photoFile;
            }
        } catch (IOException ex) {
            // Error occurred while creating the File
        }

        // Continue only if the File was successfully created
        if (photoFile != null) {
            Uri photoURI = FileProvider.getUriForFile(this,
                "com.alba.lionproject1.fileprovider", photoFile);
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
        }
    }
}

```

```

        startActivityForResult(takePictureIntent, REQUEST_TAKE_PHOTO);
    }
}
}

// 이미지 파일 만들어주는 코드
private File createImageFile() throws IOException {
    // Create an image file name
    String timeStamp = new SimpleDateFormat("yyyy_MM_dd_HH_mm_ss", Locale.KOREA).format(new Date());
    String imageFileName = "photoDate_" + timeStamp + "_";
    File storageDir = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
    File image = File.createTempFile(
        imageFileName, /* prefix */
        ".jpg", /* suffix */
        storageDir /* directory */
    );

    // Save a file: path for use with ACTION_VIEW intents
    mCurrentPhotoPath = image.getAbsolutePath();
    return image;
}

// 카메라로 촬영한 사진을 가져오는 부분
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
    super.onActivityResult(requestCode, resultCode, intent);
    try {
        if (requestCode == REQUEST_TAKE_PHOTO) {
            if (resultCode == RESULT_OK) {
                File file = new File(mCurrentPhotoPath);
                Bitmap bitmap = MediaStore.Images.Media
                    .getBitmap(getApplicationContext(), Uri.fromFile(file));
                if (bitmap != null) {
                    ExifInterface ei = new ExifInterface(mCurrentPhotoPath);
                    int orientation = ei.getAttributeInt(ExifInterface.TAG_ORIENTATION,
                        ExifInterface.ORIENTATION_UNDEFINED);

                    Bitmap rotatedBitmap;
                    switch (orientation) {

                        case ExifInterface.ORIENTATION_ROTATE_90:
                            rotatedBitmap = rotateImage(bitmap, 90);
                            break;

                        case ExifInterface.ORIENTATION_ROTATE_180:
                            rotatedBitmap = rotateImage(bitmap, 180);
                            break;

                        case ExifInterface.ORIENTATION_ROTATE_270:
                            rotatedBitmap = rotateImage(bitmap, 270);
                            break;

                        case ExifInterface.ORIENTATION_NORMAL:
                        default:
                            rotatedBitmap = bitmap;
                    }
                }
            }
        }
    }
}

// -----이미지 뷰에 수정된 사진을 띄우는 코드-----

```

```

        if (dividePath == 1) {
            imageView_at.setImageBitmap(rotatedBitmap);
        } else if (dividePath == 2) {
            imageView_after.setImageBitmap(rotatedBitmap);
        }
    }
}
} catch (Exception error) {
    error.printStackTrace();
}
}

public static Bitmap rotateImage(Bitmap source, float angle) {
    Matrix matrix = new Matrix();
    matrix.postRotate(angle);
    return Bitmap.createBitmap(source, 0, 0, source.getWidth(), source.getHeight(),
        matrix, true);
}

private static void SaveEvent(String calendardate, String month, String year) {
    DBOpenHelper dbOpenHelper = new DBOpenHelper(context);
    SQLiteDatabase database = dbOpenHelper.getWritableDatabase();
    dbOpenHelper.SaveEvent("event", "time", calendardate, month, year, database);
    dbOpenHelper.close();
}

// DB 데이터 읽어오기?
private Cursor getregisterCursor() {
    registerWorkDBHelper db = registerWorkDBHelper.getInstance(this);
    return db.getReadableDatabase().query(
        registerWorkContract.registerWorkEntry.TABLE_NAME, null, null, null, null, null,
        registerWorkContract.registerWorkEntry._ID);
}

public Bitmap rotate(Bitmap bitmap, int degrees) {
    if (degrees != 0 && bitmap != null) {
        Matrix m = new Matrix();
        m.setRotate(degrees, (float) bitmap.getWidth() / 2,
            (float) bitmap.getHeight() / 2);

        try {
            Bitmap converted = Bitmap.createBitmap(bitmap, 0, 0,
                bitmap.getWidth(), bitmap.getHeight(), m, true);
            if (bitmap != converted) {
                bitmap.recycle();
                bitmap = converted;
            }
        } catch (OutOfMemoryError ex) {
            // 메모리가 부족하여 회전을 시키지 못할 경우 그냥 원본을 반환합니다.
        }
    }
    return bitmap;
}

public int exifOrientationToDegrees(int exifOrientation) {
    if (exifOrientation == ExifInterface.ORIENTATION_ROTATE_90) {
        return 90;
    }
}

```

```

} else if (exifOrientation == ExifInterface.ORIENTATION_ROTATE_180) {
    return 180;
} else if (exifOrientation == ExifInterface.ORIENTATION_ROTATE_270) {
    return 270;
}
return 0;
}
private Cursor getTotalCursor() {
    totalDB db = totalDB.getInstance(this),
    return db.getReadableDatabase().query(
        DataBases.CreatDB._TABLENAME0, null, null, null, null, null, DataBases.CreatDB._ID);
}
}
}

```

registerWorkContract.java

```

package com.alba.lionproject1;

import android.provider.BaseColumns;

public class registerWorkContract {
    private registerWorkContract() {

    }

    public static class registerWorkEntry implements BaseColumns {
        public static final int DB_VERSION = 1;
        public static final String TABLE_NAME = "registerWork";
        public static final String COLUMN_NAME_DATE = "date";
        public static final String COLUMN_NAME_TITLE = "title";
        public static final String COLUMN_NAME_FROM = "rwfrom";
        public static final String COLUMN_NAME_TO = "rwto";
        public static final String COLUMN_NAME_MEMO = "rwmemo";
        public static final String COLUMN_NAME_ATPHOTOPATH = "path_at";
        public static final String COLUMN_NAME_AFTERPHOTOPATH = "path_after";
        public static final String COLUMN_NAME_TODATE = "todate";
        public static final String _CREATE = "CREATE TABLE IF NOT EXISTS " + TABLE_NAME + "("
            + _ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "
            + COLUMN_NAME_DATE + " TEXT NOT NULL , "
            + COLUMN_NAME_TITLE + " TEXT NOT NULL , "
            + COLUMN_NAME_FROM + " TEXT NOT NULL , "
            + COLUMN_NAME_TO + " TEXT NOT NULL , "
            + COLUMN_NAME_MEMO + " TEXT NOT NULL , "
            + COLUMN_NAME_ATPHOTOPATH + " TEXT NOT NULL, "
            + COLUMN_NAME_AFTERPHOTOPATH + " TEXT NOT NULL , "
            + COLUMN_NAME_TODATE + " TEXT NOT NULL);";
    }
}

```

registerWorkDBHelper.java

```

package com.alba.lionproject1;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class registerWorkDBHelper extends SQLiteOpenHelper {
    public static registerWorkDBHelper sInstance;

    public static final String SQL_CREATE_ENTRIES = registerWorkContract.registerWorkEntry._CREATE;
    public static final String SQL_DELETE_ENTRIES =
        "DROP TABLE IF EXISTS " + registerWorkContract.registerWorkEntry.TABLE_NAME;

    public static registerWorkDBHelper getInstance(Context context) {
        if (sInstance == null) {
            sInstance = new registerWorkDBHelper(context);
        }
        return sInstance;
    }

    public registerWorkDBHelper(@Nullable Context context) {
        super(context, registerWorkContract.registerWorkEntry.TABLE_NAME, null,
registerWorkContract.registerWorkEntry.DB_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(SQL_CREATE_ENTRIES);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL(SQL_DELETE_ENTRIES);
        onCreate(db);
    }
}

```

SelectParttime.java

```

package com.alba.lionproject1;

import androidx.appcompat.app.AppCompatActivity;
import androidx.core.content.res.ResourcesCompat;

import android.app.Activity;
import android.content.ContentValues;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Color;
import android.graphics.Typeface;
import android.graphics.drawable.Drawable;
import android.os.Bundle;
import android.view.View;
import android.view.WindowManager;

```

```

import android.widget.Button;
import android.widget.FrameLayout;
import android.widget.ImageButton;
import android.widget.LinearLayout;
import android.widget.RadioButton;
import android.widget.RadioGroup;

public class SelectParttime extends AppCompatActivity {
    int newid = 0;
    long[] id = new long[10];
    String[] totalDBArray = new String[7];
    int NumberOfButton = 0;
    public String[] parttimenames = new String[10];
    Cursor cursor;
    LinearLayout layout_ButtonPlace;
    Button button1;
    ImageButton btn_Plus;
    RadioGroup radioGroup;
    RadioButton rb_edit;
    RadioButton rb_choose;
    RadioButton rb_delete;
    String chooseOrEditOrDelete = "";
    Myapplication myapplication;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_select_parttime);
        //팝업창 형식처럼 띄우게 해주는 코드
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_select_parttime);
        myapplication = (Myapplication) getApplication();
        for (int i = 0; i < 7; i++) {
            totalDBArray[i] = "";
        }
    }

    rb_edit = (RadioButton) findViewById(R.id.rb_edit);
    rb_choose = (RadioButton) findViewById(R.id.rb_choose);
    rb_delete = (RadioButton) findViewById(R.id.rb_delete);

    PreferenceManager.setString(getApplicationContext(), "choice", "edit");
    rb_edit.setChecked(true);
    chooseOrEditOrDelete = "edit";
    PreferenceManager.setInt(getApplicationContext(), "count", 2);

    //데이터베이스 READ
    cursor = getTotalCursor();
    NumberOfButton = cursor.getCount();
    for (int i = 0; i < NumberOfButton; i++) {
        cursor.moveToPosition(i);
        parttimenames[i] = cursor.getString(1);
        id[i] = cursor.getLong(0);

    }
    newid = (int) id[id.length - 1];
    //만들때 1,2,3 이런식으로 인식됨.
}

```

```

        for (int i = 0; i < NumberOfButton; i++) {
            creatButton(parttimenames[i], id[i], i);
        }
        createPlusButton();
    }

    radioGroup = (RadioGroup) findViewById(R.id.radioGroup);

    radioGroup.setOnCheckedChangeListener((radioGroup, i) -> {
        if (i == R.id.rb_choose) {
            chooseOrEditOrDelete = "choose";
            PreferenceManager.setString(getApplicationContext(), "choice", "choose");
        } else if (i == R.id.rb_edit) {
            chooseOrEditOrDelete = "edit";
            PreferenceManager.setString(getApplicationContext(), "choice", "edit");
        } else if (i == R.id.rb_delete) {
            chooseOrEditOrDelete = "delete";
            PreferenceManager.setString(getApplicationContext(), "choice", "delete");
        }
    });
}

private void creatButton(final String nameOfButton, final long ID, final int position) //알바이름 설정 및
여러가지 설정 버튼 생성
{
    //여기는 xml이 아닌 버튼을 추가시켜주기 위하여 java내에서 화면을 설정하였습니다.
    layout_ButtonPlace = (LinearLayout) findViewById(R.id.layout);
    button1 = new Button(this);
    button1.setText(nameOfButton); //버튼 텍스트
    Typeface typeface = ResourcesCompat.getFont(getApplicationContext(), R.font.font);
    FrameLayout.LayoutParams params = new FrameLayout.LayoutParams(
        FrameLayout.LayoutParams.WRAP_CONTENT,
        FrameLayout.LayoutParams.WRAP_CONTENT);
    params.setMargins(40, 40, 40, 0);
    params.width = 450; //버튼크기
    params.height = 120; //버튼크기
    button1.setLayoutParams(params);
    button1.setTypeface(typeface);
    button1.setTextSize(20);
    button1.setBackgroundResource(R.drawable.circle);
    layout_ButtonPlace.addView(button1); //버튼추가
    setContentView(layout_ButtonPlace); //버튼추가
}

button1.setOnClickListener(view -> { // 이 버튼 클릭시 이벤트
    //데이터베이스 관련

    switch (chooseOrEditOrDelete) {
        case "edit": {
            cursor = getTotalCursor();
            cursor.moveToPosition(position);
            for (int i = 1; i <= 7; i++) {
                totalDBArray[i - 1] = cursor.getString(i);
            }
            if (totalDBArray[2].equals("")) //나중에 지워도 되는거
            {
                totalDBArray[2] = "null,null,null,null,null,null";
            }
            Intent intent = new Intent(SelectParttime.this, partTimeContent.class);

```

```

        intent.putExtra("id", ID);
        PreferenceManager.setString(getApplicationContext(), "spid", Integer.toString((int) ID));
        intent.putExtra("totaldbarray", totalDBArray);
        startActivity(intent);
        finish();
        break;
    }
    case "choose": {

        Intent intent = new Intent(SelectParttime.this, MainActivity.class);
        Activity activity = (MainActivity) MainActivity.main_activity;
        myapplication.setNameOfAlba(nameOfButton);
        PreferenceManager.setString(getApplicationContext(), "albaname", nameOfButton);
        finish();
        startActivity(intent);
        activity.finish();
        break;
    }
    case "delete": {
        PreferenceManager.setString(getApplicationContext(),"albaname","");
        totalDB totaldb = totalDB.getInstance(getApplicationContext());
        registerWorkDBHelper registerDB =
registerWorkDBHelper.getInstance(getApplicationContext());
        totaldb.getWritableDatabase().delete(DataBases.CreatDB._TABLENAME0,
DataBases.CreatDB._ID + " = " + ID, null);

        registerDB.getWritableDatabase().delete(registerWorkContract.registerWorkEntry.TABLE_NAME,
            registerWorkContract.registerWorkEntry.COLUMN_NAME_TITLE + " = ?",
new String[]{button1.getText().toString()});
        totaldb.close();
        registerDB.close();
        Intent intent1 = new Intent(SelectParttime.this,MainActivity.class);
        Intent intent = new Intent(SelectParttime.this, SelectParttime.class);
        startActivity(intent1);
        startActivity(intent);
        finish();
        break;
    }
}
});

private Cursor getTotalCursor() {
    totalDB db = totalDB.getInstance(this),
    return db.getReadableDatabase().query(
        DataBases.CreatDB._TABLENAME0, null, null, null, null, null, null, DataBases.CreatDB._ID);

}

private void createPlusButtonn()// 알바 추가 버튼 생성
{ //여기는 xml이 아닌 버튼을 추가시켜주기 위하여 java내에서 화면을 설정하였습니다.
    layout_ButtonPlace = (LinearLayout) findViewById(R.id.layout);
    btn_Plus = new ImageButton(this);
    Drawable myDrawable = getResources().getDrawable(R.drawable.plusbutton, null);
    btn_Plus.setImageDrawable(myDrawable);
    FrameLayout.LayoutParams params = new FrameLayout.LayoutParams(
        FrameLayout.LayoutParams.WRAP_CONTENT,
        FrameLayout.LayoutParams.WRAP_CONTENT);

```

```

params.setMargins(20, 20, 20, 20);
params.width = 100;
params.height = 100;
btn_Plus.setLayoutParams(params);
btn_Plus.setBackgroundColor(Color.parseColor("#fbf5df"));
layout_ButtonPlace.addView(btn_Plus);
setContentView(layout_ButtonPlace);

btn_Plus.setOnClickListener(view -> {

    ContentValues myContentValues = new ContentValues();
    myContentValues.put(DataBases.CreatDB.PARTTIMENAME, "이름 수정해");
    myContentValues.put(DataBases.CreatDB.HOURLYWAGE, "8500");
    myContentValues.put(DataBases.CreatDB.DAYOFWEEK, "null,null,null,null,null,null");
    myContentValues.put(DataBases.CreatDB.WORKINGSTART, "null");
    myContentValues.put(DataBases.CreatDB.WORKINGFINISH, "null");
    myContentValues.put(DataBases.CreatDB.NUMBER_OF_PEOPLE, "false");
    myContentValues.put(DataBases.CreatDB.MEAL_TIME, "0");
    SQLiteDatabase db = totalDB.getInstance(getApplicationContext()).getWritableDatabase();
    db.insert(DataBases.CreatDB._TABLENAME0, null, myContentValues);

    // 이버튼 클릭시 재 클릭시 화면안의 모든 요소를 지우고 다시 그려준다. 실제로 코딩할 때는
    데이터베이스에 저장된 값의 개수를 불러와서
    // 그 개수와 알바 이름들을 불러오면 될거같습니다..
    layout_ButtonPlace.removeAllViews();
    for (int i = 0; i < NumberofButton; i++) {
        creatButton(parttimenames[i], id[i], NumberofButton);
    }
    creatButton("이름 수정해", newid + 1, NumberofButton + 1);
    createPlusButton();
    Intent intent = new Intent(SelectParttime.this, SelectParttime.class);
    startActivity(intent);
    finish();
}

);
}

}

```

Timepicker.java

```

package com.alba.lionproject1;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.Intent;
import android.content.res.Configuration;
import android.os.Build;
import android.os.Bundle;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.TextView;

```

```

import android.widget.TimePicker;
import java.util.Locale;
import java.util.Objects;

public class Timepicker extends AppCompatActivity {

    TimePicker timePicker;
    public static String mins, hours;
    public static Context timecontext;
    public static String screenview;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_timepicker);
        timecontext = this;

        //한국으로 설정
        Locale locale = Locale.KOREA; //Your specific locale. Locale.setDefault(locale);
        Locale.setDefault(locale);
        Configuration configuration = timecontext.getResources().getConfiguration();
        configuration.setLocale(locale);
        configuration.setLayoutDirection(locale);

        timecontext.createConfigurationContext(configuration);
        //팝업창으로 설정
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);
        setContentView(R.layout.activity_timepicker);

        try {
            Intent intent = getIntent();
            screenview = Objects.requireNonNull(intent.getExtras()).getString("view");
        } catch (Exception ignored) {

        }

        timePicker = (TimePicker) findViewById(R.id.timepicker);
        hours = timePicker.getHour() + ":" ;
        mins = timePicker.getMinute() + ":" ;

        timePicker.setOnTimeChangedListener((timePicker, hour, min) -> {
            hours = Integer.toString(hour);
            mins = Integer.toString(min);
        });

        //저장 버튼
        Button btn_Save = (Button) findViewById(R.id.btn_tpSave);
        btn_Save.setOnClickListener(view -> {
            String time = getAmPm(Integer.parseInt(hours)) + " " + getAmPmHour(Integer.parseInt(hours)) + ":" +
            " " + getMins(mins);

            try {
                if (screenview.equals("ptcfrom")) {
                    final TextView txt_pFrom = ((partTimeContent)
                    partTimeContent.getContext()).findViewById(R.id.txt_pFrom);
                    txt_pFrom.setText(time);
                    finish();
                }
            }
        });
    }
}

```

```

    }

    if (screenview.equals("ptcto")) {
        final TextView txt_pTo = ((partTimeContent)
partTimeContent.getmContext()).findViewById(R.id.txt_pTo);
        txt_pTo.setText(time);
        finish();
    }
    if (screenview.equals("rwfrom")) {
        final TextView txt_rwfrom = ((registerWork)
registerWork.context).findViewById(R.id.txt_rwFrom);
        txt_rwfrom.setText(time);
        finish();
    }
    if (screenview.equals("rwto")) {
        final TextView txt_rwTo = ((registerWork) registerWork.context).findViewById(R.id.txt_rwTo);
        txt_rwTo.setText(time);
        finish();
    }
} catch (Exception e) {
    finish();
}
});

//취소 버튼
Button btn_Cancel = (Button) findViewById(R.id.btn_tpCancel);
btn_Cancel.setOnClickListener(view -> finish());
}

private String getAmPm(int hour) {
    if (hour >= 12)
        return "오후";
    else
        return "오전";
}

private int getAmPmHour(int hour) {
    if (hour >= 12) {
        return hour - 12;
    } else {
        return hour;
    }
}

private String getMins(String min) {
    int numMin = Integer.parseInt(min);
    if (numMin < 10) {
        return "0" + min;
    } else {
        return min;
    }
}

}

```

totalDB.java

```
package com.alba.lionproject1;

import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

import androidx.annotation.Nullable;

public class totalDB extends SQLiteOpenHelper {
    private static totalDB tdb;
    private static final String CREATE_TOTAL_INFORMATION_TABLE = DataBases.CreatDB._CREATE0;
    private static final String DROP_TOTAL_INFORMATION_TABLE = DataBases.CreatDB._DROP0;

    public static totalDB getInstance(Context context) {
        if (tdb == null) {
            tdb = new totalDB(context);
        }
        return tdb;
    }

    public totalDB(@Nullable Context context) {
        super(context, DataBases.CreatDB._TABLENAME0, null, DataBases.CreatDB.DB_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(CREATE_TOTAL_INFORMATION_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL(DROP_TOTAL_INFORMATION_TABLE);
        onCreate(db);
    }
}
```

부록-2(xml코드)

activity_get_date_pop_up.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:background="#fbf5df"

    >

    <DatePicker
        android:id="@+id/DatePicker"
        android:theme="@style/NumberPickerStyle"
        android:datePickerMode="spinner"
        android:calendarViewShown="false"
        android:layout_gravity="center_horizontal"
        android:layout_height="match_parent"
        android:layout_width="match_parent"/>

    <Button
        android:id="@+id	btn_Close"
        android:layout_width="match_parent"
        android:background="@drawable/totalbuttonshape"
        android:layout_height="match_parent"
        android:fontFamily="@font/font"
        android:onClick="mOnClick"
        android:textColor="#FFFFFF"
        android:textSize="30sp"
        android:text="저장" />

</LinearLayout>
```

activity_intro.xml

```
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@android:color/holo_orange_light"
    android:orientation="vertical"
    tools:context=".Intro">

    <TextView
        android:id="@+id/txt_extra_p"
        android:layout_width="wrap_content"
```

```

    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginStart="26dp"
    android:text="d"
    android:textColor="#FFFFFF"
    android:textSize="50sp"
    android:textStyle="bold"
    android:layout_marginLeft="26dp" />

<TextView
    android:id="@+id/txt_p"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginEnd="0dp"
    android:text="p a y   a y"
    android:textColor="#FFFFFF"
    android:textSize="50sp"
    android:textStyle="bold"
    android:layout_marginRight="0dp" />

</FrameLayout>

```

activity_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#fbf5df"
    android:focusedByDefault="true"
    tools:context=".MainActivity">

    <ScrollView
        android:id="@+id	scroll"
        android:layout_width="0dp"
        android:layout_height="0dp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <LinearLayout
            android:id="@+id/sclinear"
            android:layout_width="match_parent"
            android:layout_height="match_parent">

            <TableLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_gravity="start">

```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="@font/font"
        android:text="알바계산기"
        android:textColor="@color/black"
        android:textSize="30sp" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center_horizontal|center_vertical"
    android:layout_weight="1"
    android:gravity="center_horizontal|center_vertical">

    <Button
        android:id="@+id/btn_spFrom"
        android:layout_width="52dp"
        android:layout_height="match_parent"
        android:layout_weight="5"
        android:background="@drawable/totalbuttonshape"
        android:fontFamily="@font/font"
        android:gravity="center"
        android:text="몇 일부터"
        android:textColor="#FFFFFF"
        android:textSize="12sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/txt_from"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="40"
        android:fontFamily="@font/font"
        android:gravity="center"
        android:text=""
        android:textAlignment="center"
        android:textSize="16sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/txt_fromto"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="10"
        android:text="@string/fromTo"
        android:textAlignment="center"
        android:textSize="30sp"
        android:textStyle="bold"
        android:gravity="center_horizontal" />
```

```
<Button  
    android:id="@+id/btn_spTo"  
    android:layout_width="52dp"  
    android:layout_height="wrap_content"  
    android:layout_weight="5"  
    android:background="@drawable/totalbuttonshape"  
    android:fontFamily="@font/font"  
    android:gravity="center"  
    android:text="몇일까지"  
    android:textColor="#FFFFFF"  
    android:textSize="12sp"  
    android:textStyle="bold" />  
  
<TextView  
    android:id="@+id/txt_to"  
    android:layout_width="wrap_content"  
    android:layout_height="match_parent"  
    android:layout_weight="40"  
    android:fontFamily="@font/font"  
    android:gravity="center"  
    android:text=""  
    android:textAlignment="center"  
    android:textSize="16sp"  
    android:textStyle="bold" />  
</LinearLayout>  
  
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center_horizontal|center_vertical"  
    android:layout_weight="30"  
    android:gravity="start"  
    android:orientation="horizontal">  
  
<CheckBox  
    android:id="@+id/cb_susoup"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:buttonTint="#ED7E76"  
    android:layout_margin="10dp"  
    android:fontFamily="@font/font"  
    android:text="수습기간이면 체크해"  
    android:textSize="16sp" />  
  
<ImageButton  
    android:id="@+id/btn_helpProbationPeriod"  
    android:layout_width="30dp"  
    android:layout_height="30dp"  
    android:layout_margin="10dp"  
    android:background="@drawable/questionmark"  
    android:textSize="5sp"  
    android:contentDescription="@string/todo" />  
  
</LinearLayout>  
  
<LinearLayout  
    android:layout_width="match_parent"
```

```
        android:layout_height="250dp"
        android:layout_gravity="center_horizontal|center_vertical"
        android:layout_weight="30"
        android:gravity="center_horizontal|center_vertical">

    <EditText
        android:id="@+id/txt_monthlyPay"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_margin="20dp"
        android:layout_weight="1"
        android:background="@drawable/border"
        android:ems="10"
        android:focusable="false"
        android:fontFamily="@font/font"
        android:inputType="textPersonName"
        android:text="0 원"
        android:textAlignment="center"
        android:textSize="50sp"
        android:textStyle="bold"
        android:gravity="center" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal|center_vertical"
    android:layout_weight="30"
    android:gravity="center_horizontal|center_vertical">

    <Button
        android:id="@+id	btn_ChooseParttime"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_margin="15dp"
        android:layout_weight="1"
        android:background="@drawable/totalbuttonshape"
        android:fontFamily="@font/font"
        android:text="@string/chooseparttime"
        android:textColor="#fffff"
        android:textSize="30sp" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_gravity="center_horizontal|center_vertical"
    android:layout_weight="50"
    android:gravity="center_horizontal|center_vertical">

    <com.alba.lionproject1.customcalendarview
        android:id="@+id/customcalendar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

    </com.alba.lionproject1.customcalendarview>
</LinearLayout>
```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:orientation="horizontal">

    <Button
        android:id="@+id/btn_EvidenceCsv"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="18dp"
        android:layout_marginEnd="9dp"
        android:layout_weight="1"
        android:background="@drawable/totalbuttonshape"
        android:fontFamily="@font/font"
        android:text="증거물 파일 생성"
        android:textColor="#FFFFFF"
        android:textSize="25sp"
        android:layout_marginLeft="18dp"
        android:layout_marginRight="9dp" />

    <Button
        android:id="@+id/btn_ImgFilePath"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="9dp"
        android:layout_marginEnd="18dp"
        android:layout_weight="0.5"
        android:background="@drawable/totalbuttonshape"
        android:fontFamily="@font/font"
        android:text="파일 경로"
        android:textColor="#FFFFFF"
        android:textSize="25sp"
        android:layout_marginLeft="9dp"
        android:layout_marginRight="18dp" />

</LinearLayout>
</TableLayout>
</LinearLayout>
</ScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>

```

activity_part_time_content.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/layout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="start"
    android:background="#fbf5df"
    android:gravity="start"
    android:orientation="vertical" >

    <EditText

```

```
    android:id="@+id/txt_ParttimeName"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="@font/font"
    android:background="@drawable/buttonline"
    android:textSize="20sp"
    android:ems="10"
    android:gravity="center"
    android:inputType="textPersonName"
    android:text="알바이름을쓰시오" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:layout_weight="1.5"
        android:gravity="center"
        android:textSize="20sp"
        android:fontFamily="@font/font"
        android:text="기본시급"
        android:textAlignment="center" />

    <EditText
        android:id="@+id/txt_hourlypay"
        android:layout_width="125dp"
        android:layout_height="49dp"
        android:textSize="24sp"
        android:layout_weight="2"
        android:background="@drawable/buttonline"
        android:fontFamily="@font/font"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="7500" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fontFamily="@font/font"
    android:gravity="center"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/textView9"
        android:layout_width="wrap_content"
        android:background="@drawable/smalltitles"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textColor="#fffff"
        android:gravity="center"
        android:textSize="20sp"
        android:fontFamily="@font/font"
        android:text="고정근무요일" />
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <CheckBox
        android:id="@+id/cb_mon"
        android:layout_width="wrap_content"
        android:fontFamily="@font/font"
        android:textSize="20sp"
        android:layout_height="wrap_content"
        android:buttonTint="#ED7E76"
        android:layout_weight="1.5"
        android:text="@string/monday" />

    <CheckBox
        android:id="@+id/cb_tues"
        android:layout_width="wrap_content"
        android:textSize="20sp"
        android:buttonTint="#ED7E76"
        android:layout_height="wrap_content"
        android:fontFamily="@font/font"
        android:layout_weight="1.5"
        android:text="@string/tuesday" />

    <CheckBox
        android:id="@+id/cb_wed"
        android:layout_width="wrap_content"
        android:buttonTint="#ED7E76"
        android:textSize="20sp"
        android:layout_height="wrap_content"
        android:layout_weight="1.5"
        android:fontFamily="@font/font"
        android:text="@string/wednesday" />

    <CheckBox
        android:id="@+id/cb_thurs"
        android:layout_width="wrap_content"
        android:buttonTint="#ED7E76"
        android:textSize="20sp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:fontFamily="@font/font"
        android:text="@string/thursday" />

    <CheckBox
        android:id="@+id/cb_fri"
        android:layout_width="wrap_content"
        android:textSize="20sp"
        android:buttonTint="#ED7E76"
        android:fontFamily="@font/font"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/friday" />

    <CheckBox
        android:id="@+id/cb_sat"
```

```
        android:layout_width="wrap_content"
        android:textSize="20sp"
        android:buttonTint="#ED7E76"
        android:layout_height="wrap_content"
        android:fontFamily="@font/font"
        android:layout_weight="1"
        android:text="@string/saturday" />

    <CheckBox
        android:id="@+id/cb_sun"
        android:layout_width="wrap_content"
        android:textSize="20sp"
        android:buttonTint="#ED7E76"
        android:fontFamily="@font/font"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/sunday" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/textView8"
        android:textSize="20sp"
        android:layout_width="wrap_content"
        android:fontFamily="@font/font"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:textColor="#ffffffff"
        android:gravity="center"
        android:background="@drawable/smalltitles"
        android:text="고정근무시간" />
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <Button
        android:id="@+id	btn_tpFrom"
        android:layout_width="50dp"
        android:background="@drawable/totalbuttonshape"
        android:layout_height="wrap_content"
        android:text="@string/fromtime"
        android:textColor="#ffffffff"
        android:layout_gravity="center"
        android:gravity="center"
        android:fontFamily="@font/font"
        />

    <TextView
        android:id="@+id/txt_pFrom"
        android:layout_width="70dp"
        android:fontFamily="@font/font"
        android:layout_height="wrap_content" />
```

```
<TextView
    android:id="@+id/textView7"
    android:layout_width="70dp"
    android:fontFamily="@font/font"
    android:layout_height="match_parent"
    android:gravity="center"
    android:text="~"
    android:textSize="24sp" />

<Button
    android:id="@+id/btn_tpTo"
    android:background="@drawable/totalbuttonshape"
    android:fontFamily="@font/font"
    android:layout_width="50dp"
    android:textColor="#ffffffff"
    android:layout_height="wrap_content"
    android:text="@string/fromto" />

<TextView
    android:id="@+id/txt_pTo"
    android:layout_width="70dp"
    android:fontFamily="@font/font"
    android:layout_height="wrap_content"
    />

</LinearLayout>

<TextView
    android:id="@+id/textView10"
    android:layout_width="match_parent"
    android:fontFamily="@font/font"
    android:background="@drawable/smalltitles"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:textColor="#ffffffff"
    android:gravity="center"
    android:text="5인 이상 사업장 입니까?" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

<RadioGroup
    android:id="@+id/rb_group"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="9"
    android:orientation="horizontal">

<RadioButton
    android:id="@+id/rb_yes"
    android:textSize="20sp"
    android:layout_width="wrap_content"
    android:fontFamily="@font/font"
    android:layout_height="match_parent"
    android:layout_marginStart="60dp"
    android:text="@string/yes" />
```

```
<RadioButton
    android:id="@+id/rb_no"
    android:layout_width="wrap_content"
    android:textSize="20sp"
    android:fontFamily="@font/font"
    android:layout_height="match_parent"
    android:layout_marginStart="60dp"
    android:text="@string/no" />

</RadioGroup>

<ImageButton
    android:id="@+id/btn_helpWorkplace"
    android:layout_width="25dp"
    android:layout_height="30dp"
    android:layout_margin="10dp"
    android:layout_weight="1"
    android:background="@drawable/questionmark"
    android:textSize="20sp"
    android:contentDescription="@string/todo" />

</LinearLayout>

<TextView
    android:id="@+id/text"
    android:layout_width="match_parent"
    android:fontFamily="@font/font"
    android:background="@drawable/smalltitles"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:textColor="#ffffff"
    android:gravity="center"
    android:text="무급 휴식시간(식사시간)" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="60"
        android:orientation="horizontal">
        <EditText
            android:id="@+id/txt_mealtime"
            android:layout_width="200dp"
            android:layout_height="match_parent"
            android:background="@drawable/buttonline"
            android:layout_weight="60" />
        <TextView
            android:layout_marginStart="15dp"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:fontFamily="@font/font"
            android:textSize="35sp" />
    
```

```

        android:text="분"
    />
</LinearLayout>
<LinearLayout
    android:layout_weight="2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <ImageButton
        android:layout_margin="10dp"
        android:id="@+id/btn_helpBreaktime"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:background="@drawable/questionmark"
        android:textSize="20sp"
        android:contentDescription="@string/todo" />
</LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <Button
        android:id="@+id/btn_pSave"
        android:layout_width="wrap_content"
        android:textSize="20sp"
        android:layout_height="wrap_content"
        android:background="@drawable/totalbuttonshape"
        android:textColor="#ffffffff"
        android:fontFamily="@font/font"
        android:layout_margin="10dp"
        android:layout_weight="1"
        android:text="저장" />

    <Button
        android:id="@+id/btn_pCancel"
        android:layout_width="wrap_content"
        android:textColor="#ffffffff"
        android:layout_height="wrap_content"
        android:fontFamily="@font/font"
        android:textSize="20sp"
        android:background="@drawable/totalbuttonshape"
        android:layout_margin="10dp"
        android:layout_weight="1"
        android:text="취소" />
</LinearLayout>
</LinearLayout>

```

activity_register_work.xml

```

<?xml version="1.0" encoding="utf-8"?>

<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"

```

```
xmlns:tools="http://schemas.android.com/tools"
    android:background="#fbf5df"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:textAlignment="center"
    android:gravity="center_horizontal">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        android:layout_margin="10dp"
        android:gravity="center"
        android:orientation="vertical"
        tools:context=".registerWork">

        <TextView
            android:id="@+id/txt_date"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:fontFamily="@font/font"
            android:text="(날짜텍스트 ex.2020.03.25)" />

        <TextView
            android:id="@+id/albaName"
            android:layout_width="match_parent"
            android:fontFamily="@font/font"
            android:layout_height="wrap_content"
            android:layout_margin="20dp"
            android:background="@null"
            android:focusable="false"
            android:gravity="center"
            android:text="알바이름입력"
            android:textSize="50sp" />

        <TextView
            android:id="@+id/textView14"
            android:textColor="@color/black"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textSize="20sp"
            android:fontFamily="@font/font"
            android:gravity="center"
            android:text="오늘근무시간" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="60dp"
        android:orientation="horizontal">

        <Button
            android:id="@+id/btn_rwDateFrom"
            android:layout_width="50dp"
            android:layout_height="50dp"
            android:textColor="#ffffffff"
            android:textSize="17sp"
            android:background="@drawable/totalbuttonshape"
            android:fontFamily="@font/font"
            android:gravity="center"
```

```
    android:layout_gravity="center"
    android:layout_weight="0.5"
    android:text="현재날짜" />

<TextView
    android:id="@+id/txt_rwDateFrom"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:textSize="20sp"
    android:layout_gravity="center"
    android:layout_weight="1"
    android:text="@string/blanks"
    android:fontFamily="@font/font"
    android:textColor="@color/black"
    android:gravity="center" />

<TextView
    android:id="@+id/textView123"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:fontFamily="@font/font"
    android:layout_gravity="center"
    android:textColor="@color/black"
    android:layout_weight="0.8"
    android:gravity="center"
    android:text="@string/fromTo"
    android:textSize="30sp" />

<Button
    android:id="@+id/btn_rwDateTo"
    android:background="@drawable/totalbuttonshape"
    android:layout_width="50dp"
    android:textSize="17sp"
    android:fontFamily="@font/font"
    android:gravity="center"
    android:layout_gravity="center"
    android:layout_height="50dp"
    android:textColor="#ffffffff"
    android:layout_weight="0.5"
    android:text="몇일까지" />

<TextView
    android:id="@+id/txt_rwDateTo"
    android:fontFamily="@font/font"
    android:textSize="20sp"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="center"
    android:layout_weight="1"
    android:textColor="@color/black"
    android:text="@string/blanks"
    android:gravity="center" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
```

```
    android:layout_height="60dp"
    android:orientation="horizontal">>

    <Button
        android:id="@+id/btn_rwFrom"
        android:layout_width="50dp"
        android:layout_height="50dp"
        android:textColor="#ffffffff"
        android:textSize="17sp"
        android:background="@drawable/totalbuttonshape"
        android:fontFamily="@font/font"
        android:gravity="center"
        android:layout_gravity="center"
        android:layout_weight="0.5"
        android:text="@string/towhattime" />

    <TextView
        android:id="@+id/txt_rwFrom"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:textSize="20sp"
        android:layout_gravity="center"
        android:layout_weight="1"
        android:text="@string/blanks"
        android:fontFamily="@font/font"
        android:textColor="@color/black"
        android:gravity="center" />

    <TextView
        android:id="@+id/textView12"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:fontFamily="@font/font"
        android:layout_gravity="center"
        android:textColor="@color/black"
        android:layout_weight="0.8"
        android:gravity="center"
        android:text="@string/fromTo"
        android:textSize="30sp" />

    <Button
        android:id="@+id/btn_rwTo"
        android:background="@drawable/totalbuttonshape"
        android:layout_width="50dp"
        android:textSize="17sp"
        android:fontFamily="@font/font"
        android:gravity="center"
        android:layout_gravity="center"
        android:layout_height="50dp"
        android:textColor="#ffffffff"
        android:layout_weight="0.5"
        android:text="@string/fromwhattime" />

    <TextView
        android:id="@+id/txt_rwTo"
        android:fontFamily="@font/font"
        android:textSize="20sp"
        android:layout_width="wrap_content"
        android:layout_height="match_parent"
```

```
    android:layout_gravity="center"
    android:layout_weight="1"
    android:textColor="@color/black"
    android:text="@string/blanks"
    android:gravity="center" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

<TextView
    android:id="@+id/txt_memo222"
    android:layout_width="wrap_content"
    android:textSize="20sp"
    android:textColor="@color/black"
    android:fontFamily="@font/font"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center"
    android:text="메모" />

</LinearLayout>

<com.google.android.material.textfield.TextInputLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/txt_memo"
    android:layout_width="match_parent"
    android:fontFamily="@font/font"
    android:layout_height="wrap_content"
    android:background="@drawable/buttonline"
    android:hint="@null"
    android:textSize="24sp" />
</com.google.android.material.textfield.TextInputLayout>

<TextView
    android:id="@+id/textView15"
    android:textColor="@color/black"
    android:fontFamily="@font/font"
    android:textSize="20sp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="출근인증사진" />

<ImageView
    android:id="@+id/iv_picture_at_work"
    android:layout_width="match_parent"
    android:layout_height="271dp"
    app:srcCompat="@drawable/border"
    android:contentDescription="@string/todo"
/>
```

```
<Button
    android:id="@+id	btn_takepicture"
    android:background="@drawable/totalbuttonshape"
    android:fontFamily="@font/font"
    android:textColor="#fffff"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/TakePictureSaying" />
<TextView
    android:id="@+id	txt_gettingoffthework"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/photo_getoffwork"
    android:textColor="#000000"
    android:fontFamily="@font/font"
    android:textSize="20sp"

    />
<ImageView
    android:id="@+id/iv_picture_out_work"
    android:layout_width="match_parent"
    android:layout_height="271dp"
    app:srcCompat="@drawable/border"
    android:contentDescription="@string/todo" />

<Button
    android:id="@+id	btn_takepicture_afterWork"
    android:background="@drawable/totalbuttonshape"
    android:fontFamily="@font/font"
    android:textColor="#fffff"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/TakePictureSaying" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="bottom|center"
    android:orientation="horizontal">

    <Button
        android:id="@+id	btn_rwSave"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="@drawable/totalbuttonshape"
        android:fontFamily="@font/font"
        android:layout_margin="10dp"
        android:layout_weight="1"
        android:textSize="20sp"
        android:textColor="#fffff"
        android:text="@string/save" />

    <Button
        android:id="@+id	btn_rwCancel"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:fontFamily="@font/font"
```

```
        android:background="@drawable/totalbuttonshape"
        android:layout_margin="10dp"
        android:layout_weight="1"
        android:textColor="#ffffffff"
        android:text="취소" />

    </LinearLayout>
</LinearLayout>
</ScrollView>
```

activity_select_parttime.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/layout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="start"
    android:background="#fbf5df"
    android:gravity="center"
    android:orientation="vertical">

    <RadioGroup
        android:layout_margin="5dp"
        android:id="@+id/radioGroup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:orientation="horizontal">

        <RadioButton
            android:id="@+id/rb_choose"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:fontFamily="@font/font"
            android:layout_weight="1"
            android:buttonTint="@color/peach"
            android:text="@string/choose" />

        <RadioButton
            android:id="@+id/rb_edit"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:buttonTint="@color/peach"
            android:layout_weight="1"
            android:text="@string/edit"
            android:layout_gravity="center"
            android:fontFamily="@font/font"/>

        <RadioButton
            android:id="@+id/rb_delete"
            android:layout_width="wrap_content"
            android:fontFamily="@font/font"
```

```
    android:buttonTint="@color/peach"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_weight="1"
    android:text="@string/delete" />

  </RadioGroup>
</LinearLayout>
```

activity_timepicker.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/timelayout"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_gravity="start"
  android:background="#fbf5df"
  android:gravity="start"
  android:orientation="vertical">

  <TimePicker
    android:layout_width="wrap_content"
    android:id="@+id/timepicker"
    android:theme="@style/NumberPickerStyle"
    android:layout_height="wrap_content"
    android:timePickerMode="spinner"/>

  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal">

    <Button
      android:id="@+id/btn_tpSave"
      android:layout_width="30dp"
      android:layout_height="50dp"
      android:layout_margin="10dp"
      android:textSize="20sp"
      android:fontFamily="@font/font"
      android:background="@drawable/totalbuttonshape"
      android:textColor="#ffffffff"
      android:layout_weight="1"
      android:text="저장" />

    <Button
      android:id="@+id/btn_tpCancel"
      android:layout_width="30dp"
      android:layout_height="50dp"
      android:textSize="20sp"
      android:background="@drawable/totalbuttonshape"
      android:fontFamily="@font/font"
      android:textColor="#ffffffff"
      android:layout_margin="10dp"
      android:layout_weight="1"
```

```
        android:text="취소" />
    </LinearLayout>
</LinearLayout>
```

alertdialog_addxls.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:background="#fbf5df"
    tools:context=".AlertDialog_filepath"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/guid"
            android:gravity="center"
            android:textSize="30sp"
            android:layout_marginBottom="30dp"
            android:fontFamily="@font/font"
            android:background="@drawable/smalltitles"/>

        <TextView
            android:id="@+id/txt_XlsExplain"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="30dp"
            android:textSize="20sp"
            android:fontFamily="@font/font" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="@string/wannaCreateaFile"
            android:gravity="center"
            android:textSize="30sp"
            android:layout_marginBottom="30dp"
            android:fontFamily="@font/font"
            android:background="@drawable/smalltitles"/>

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="bottom|end"
            android:gravity="center"
            android:orientation="horizontal">
```

```

<Button
    android:id="@+id/btn_XIsYes"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="25dp"
    android:background="@drawable/totalbuttonshape"
    android:fontFamily="@font/font"
    android:text="@string/yes"
    android:textColor="#fffff"
    android:textSize="25sp"
    android:layout_marginEnd="25dp" />

<Button
    android:id="@+id/btn_XIsNo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="25dp"
    android:background="@drawable/totalbuttonshape"
    android:fontFamily="@font/font"
    android:text="@string/no"
    android:textColor="#fffff"
    android:textSize="25sp"
    android:layout_marginStart="25dp" />
</LinearLayout>
</LinearLayout>
</ScrollView>

```

alertdialog_filepath.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:background="#fbf5df"
    tools:context=".AlertDialog_filepath"
    android:orientation="vertical">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="이미지 파일 저장 경로"
            android:gravity="center"
            android:textSize="30sp"
            android:layout_marginBottom="30dp"
            android:fontFamily="@font/font"
            android:background="@drawable/smalltitles" />

```

```

<TextView
    android:id="@+id/txt_filepath"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:fontFamily="@font/font" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="엑셀 파일 저장 경로"
    android:gravity="center"
    android:textSize="30sp"
    android:layout_marginTop="30dp"
    android:layout_marginBottom="30dp"
    android:fontFamily="@font/font"
    android:background="@drawable/smalltitles"/>

<TextView
    android:id="@+id/xlsfilepath"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="20sp"
    android:fontFamily="@font/font" />

<Button
    android:id="@+id	btn_Close"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="30dp"
    android:background="@drawable/totalbuttonshape"
    android:fontFamily="@font/font"
    android:text="닫기"
    android:textColor="#fffff"
    android:textSize="25sp" />

</LinearLayout>

</ScrollView>

```

calendar_layout.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/activity_custom_calendar"
    android:orientation="vertical"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@android:color/holo_orange_light"
        android:orientation="horizontal">

```

```
    android:paddingTop="8dp"
    android:paddingBottom="8dp">

    <ImageButton
        android:id="@+id/previousBtn"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_margin="10dp"
        android:background="@drawable/left_direction" />

    <TextView
        android:id="@+id/current_Date"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_margin="4dp"
        android:layout_weight="3"
        android:fontFamily="@font/font"
        android:gravity="center"
        android:text="@string/currendate"
        android:textColor="@android:color/white"
        android:textSize="30sp" />

    <ImageButton
        android:id="@+id/nextBtn"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_margin="10dp"
        android:background="@drawable/right_direction"
        android:contentDescription="@string/todo" />

</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingTop="4dp"
    android:paddingBottom="4dp"
    android:background="#fef691">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="@string/sunday"
        android:fontFamily="@font/font"
        android:background="#fef691"
        android:layout_marginVertical="4dp"
        android:layout_weight="1"
        android:textStyle="bold"

        />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="@string/monday"
        android:fontFamily="@font/font"
```

```
        android:background="#fef691"
        android:layout_marginVertical="4dp"
        android:layout_weight="1"
        android:textStyle="bold"

    />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/tuesday"
    android:background="#fef691"
    android:fontFamily="@font/font"
    android:layout_marginVertical="4dp"
    android:layout_weight="1"
    android:textStyle="bold"

    />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/wednesday"
    android:background="#fef691"
    android:fontFamily="@font/font"
    android:layout_marginVertical="4dp"
    android:layout_weight="1"
    android:textStyle="bold"

    />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/thursday"
    android:background="#fef691"
    android:fontFamily="@font/font"
    android:layout_marginVertical="4dp"
    android:layout_weight="1"
    android:textStyle="bold"

    />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/friday"
    android:background="#fef691"
    android:fontFamily="@font/font"
    android:layout_marginVertical="4dp"
    android:layout_weight="1"
    android:textStyle="bold"

    />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/saturday"
```

```
        android:background="#fef691"
        android:fontFamily="@font/font"
        android:layout_marginVertical="4dp"
        android:layout_weight="1"
        android:textStyle="bold"
    />

</LinearLayout>

<GridView
    android:id="@+id/gridview"
    android:layout_width="match_parent"
    android:layout_height="450dp"
    android:numColumns="7" />

</LinearLayout>
```

part_time_content_help_breaktime.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="300dp"
    android:layout_height="300dp"
    android:background="#fbf5df"
    android:orientation="vertical"
    tools:context=".partTimeContent_help_Breaktime">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="20dp"
            android:background="@drawable/smalltitles"
            android:fontFamily="@font/font"
            android:gravity="center"
            android:text="무급휴식시간이란?"
            android:textSize="30dp" />

        <TextView
            android:id="@+id/txt_help_breaktime"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="20dp"
            android:fontFamily="@font/font"
            android:textSize="20dp" />

        <Button
            android:id="@+id/btn_help_breaktime_ok"
            android:layout_width="wrap_content"
```

```

        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:background="@drawable/totalbuttonshape"
        android:text="확인"
        android:textColor="#ffffffff"
        android:textSize="20dp" />
    </LinearLayout>
</ScrollView>

```

part_time_content_help_workplace.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="300dp"
    android:layout_height="300dp"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:background="#fbf5df"
    android:orientation="vertical"
    tools:context=".partTimeContent_help_Workplace">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="20dp"
            android:background="@drawable/smalltitles"
            android:fontFamily="@font/font"
            android:gravity="center"
            android:text="5인 사업장이란?"
            android:textSize="30dp" />

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:fontFamily="@font/font"
            android:gravity="center"
            android:text="근로기준법 적용범위 이해를 위한 표입니다."
            android:textSize="20dp"/>

        <TableLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_margin="8dp"
            android:layout_gravity="center"
            app:layout_constraintLeft_toLeftOf = "parent"
            app:layout_constraintRight_toRightOf = "parent"
            app:layout_constraintTop_toTopOf = "parent">

            <TableRow
                android:background="#9A9494">

```

```
<TextView
    android:layout_width="90dp"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textSize="12sp"
    android:background="@drawable/smalltitles"
    android:textColor="#FFFFFF"
    android:textStyle="bold"
    android:text="구분"/>
<TextView
    android:layout_width="90dp"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textSize="12sp"
    android:background="@drawable/smalltitles"
    android:textColor="#FFFFFF"
    android:textStyle="bold"
    android:text="5인 미만 사업장"/>
<TextView
    android:layout_width="90dp"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textSize="12sp"
    android:background="@drawable/smalltitles"
    android:textColor="#FFFFFF"
    android:textStyle="bold"
    android:text="5인 이상 사업장"/>
</TableRow>
<TableRow
    android:background="#9A9494">
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:textStyle="bold"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:text="근로계약"/>
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:text="해당"/>
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:layout_marginBottom="1dp"
```

```
        android:text="해당"/>
    </TableRow>
    <TableRow
        android:background="#9A9494">
        <TextView
            android:layout_width="90dp"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textSize="12sp"
            android:background="#FFFFFF"
            android:textColor="#000000"
            android:textStyle="bold"
            android:layout_marginRight="1dp"
            android:layout_marginBottom="1dp"
            android:text="주휴수당"
            android:layout_marginEnd="1dp" />
        <TextView
            android:layout_width="90dp"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textSize="12sp"
            android:background="#FFFFFF"
            android:textColor="#000000"
            android:layout_marginRight="1dp"
            android:layout_marginBottom="1dp"
            android:text="해당"
            android:layout_marginEnd="1dp" />
        <TextView
            android:layout_width="90dp"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textSize="12sp"
            android:background="#FFFFFF"
            android:textColor="#000000"
            android:layout_marginBottom="1dp"
            android:text="해당"/>
    </TableRow>
    <TableRow
        android:background="#9A9494">
        <TextView
            android:layout_width="90dp"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textSize="12sp"
            android:background="#FFFFFF"
            android:textColor="#000000"
            android:layout_marginRight="1dp"
            android:layout_marginBottom="1dp"
            android:text="기타 유급휴일"
            android:layout_marginEnd="1dp" />
        <TextView
            android:layout_width="90dp"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textSize="12sp"
            android:background="#DDD5D5"
            android:textColor="#000000"
            android:layout_marginRight="1dp"
```

```
        android:layout_marginBottom="1dp"
        android:text="미해당"
        android:layout_marginEnd="1dp" />
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:layout_marginBottom="1dp"
        android:text="해당"/>
    </TableRow>
    <TableRow
        android:background="#9A9494">
        <TextView
            android:layout_width="90dp"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textSize="12sp"
            android:background="#FFFFFF"
            android:textColor="#000000"
            android:textStyle="bold"
            android:layout_marginRight="1dp"
            android:layout_marginBottom="1dp"
            android:text="유급휴가(연차)"
            android:layout_marginEnd="1dp" />
        <TextView
            android:layout_width="90dp"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textSize="12sp"
            android:background="#DDD5D5"
            android:textColor="#000000"
            android:layout_marginRight="1dp"
            android:layout_marginBottom="1dp"
            android:text="미해당"
            android:layout_marginEnd="1dp" />
        <TextView
            android:layout_width="90dp"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textSize="12sp"
            android:background="#FFFFFF"
            android:textColor="#000000"
            android:layout_marginBottom="1dp"
            android:text="해당"/>
    </TableRow>
    <TableRow
        android:background="#9A9494">
        <TextView
            android:layout_width="90dp"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:textSize="12sp"
            android:background="#FFFFFF"
            android:textColor="#000000"
            android:layout_marginBottom="1dp"
            android:text="해당"/>
```

```
        android:layout_marginBottom="1dp"
        android:text="해고예고"
        android:layout_marginEnd="1dp" />
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:text="해당"
        android:layout_marginEnd="1dp" />
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:layout_marginBottom="1dp"
        android:text="해당"/>
</TableRow>
<TableRow
    android:background="#9A9494">
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:textStyle="bold"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:text="휴업수당"
        android:layout_marginEnd="1dp" />
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#DDD5D5"
        android:textColor="#000000"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:text="미해당"/>
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:layout_marginBottom="1dp"
        android:text="해당"/>
</TableRow>
<TableRow
```

```
    android:background="#9A9494">
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:textStyle="bold"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:text="휴게시간"
        android:layout_marginEnd="1dp" />
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:text="해당"
        android:layout_marginEnd="1dp" />
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:layout_marginBottom="1dp"
        android:text="해당"/>
</TableRow>
<TableRow
    android:background="#9A9494">
    <TextView
        android:layout_width="90dp"
        android:layout_height="match_parent"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:background="#FFFFFF"
        android:gravity="center"
        android:text="연장/야간/휴일근로"
        android:textColor="#000000"
        android:textSize="10sp"
        android:textStyle="bold"
        android:layout_marginEnd="1dp" />
    <TextView
        android:layout_width="90dp"
        android:layout_height="35dp"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#DDD5D5"
        android:textColor="#000000"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:text="미해당"
        android:layout_marginEnd="1dp" />
```

```
<TextView
    android:layout_width="90dp"
    android:layout_height="35dp"
    android:gravity="center"
    android:textSize="12sp"
    android:background="#FFFFFF"
    android:textColor="#000000"
    android:layout_marginBottom="1dp"
    android:text="해당"/>
</TableRow>
<TableRow
    android:background="#9A9494">
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:textStyle="bold"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:text="퇴직금"
        android:layout_marginEnd="1dp" />
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:text="해당"
        android:layout_marginEnd="1dp" />
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:layout_marginBottom="1dp"
        android:text="해당"
        android:layout_marginEnd="1dp" />
</TableRow>
<TableRow
    android:background="#9A9494">
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:layout_marginBottom="1dp"
        android:text="최저임금"
        android:layout_marginEnd="1dp" />
```

```
<TextView
    android:layout_width="90dp"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textSize="12sp"
    android:background="#FFFFFF"
    android:textColor="#000000"
    android:layout_marginRight="1dp"
    android:layout_marginBottom="1dp"
    android:text="해당"
    android:layout_marginEnd="1dp" />
<TextView
    android:layout_width="90dp"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textSize="12sp"
    android:background="#FFFFFF"
    android:textColor="#000000"
    android:layout_marginBottom="1dp"
    android:text="해당"/>
</TableRow>
<TableRow
    android:background="#9A9494">
<TextView
    android:layout_width="90dp"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textSize="12sp"
    android:background="#FFFFFF"
    android:textColor="#000000"
    android:textStyle="bold"
    android:layout_marginRight="1dp"
    android:layout_marginBottom="1dp"
    android:text="생리휴가"
    android:layout_marginEnd="1dp" />
<TextView
    android:layout_width="90dp"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textSize="12sp"
    android:background="#DDD5D5"
    android:textColor="#000000"
    android:layout_marginRight="1dp"
    android:layout_marginBottom="1dp"
    android:text="미해당"
    android:layout_marginEnd="1dp" />
<TextView
    android:layout_width="90dp"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:textSize="12sp"
    android:background="#FFFFFF"
    android:textColor="#000000"
    android:layout_marginBottom="1dp"
    android:text="해당"/>
</TableRow>
<TableRow
    android:background="#9A9494">
<TextView
```

```
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#FFFFFF"
        android:textColor="#000000"
        android:textStyle="bold"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:text="취업규칙"
        android:layout_marginEnd="1dp" />
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#DDD5D5"
        android:textColor="#000000"
        android:layout_marginRight="1dp"
        android:layout_marginBottom="1dp"
        android:text="미해당"
        android:layout_marginEnd="1dp" />
    <TextView
        android:layout_width="90dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="12sp"
        android:background="#DDD5D5"
        android:textColor="#000000"
        android:layout_marginBottom="1dp"
        android:text="미해당"/>
</TableRow>
</TableLayout>

<TextView
    android:id="@+id/txt_help_workplace"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"
    android:fontFamily="@font/font"
    android:textSize="20dp" />

<Button
    android:id="@+id	btn_help_workplace_ok"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/totalbuttonshape"
    android:text="확인"
    android:textColor="#fffff"
    android:textSize="20dp" />
</LinearLayout>
</ScrollView>
```

probation_period_help.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="300dp"
    android:layout_height="300dp"
    xmlns:tools="http://schemas.android.com/tools"
    android:background="#fbf5df"
    android:focusedByDefault="true"
    android:orientation="vertical"
    tools:context=".partTimeContent_help_Breaktime">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="20dp"
            android:background="@drawable/smalltitles"
            android:fontFamily="@font/font"
            android:gravity="center"
            android:text="수습기간이란?"
            android:textSize="30dp"/>

        <TextView
            android:id="@+id/txt_probation_period_mean"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="20dp"
            android:fontFamily="@font/font"
            android:textSize="20dp"/>

        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="20dp"
            android:background="@drawable/smalltitles"
            android:fontFamily="@font/font"
            android:gravity="center"
            android:text="수습기간 포함 시 월급 계산법"
            android:textSize="30dp"/>

        <TextView
            android:id="@+id/txt_probation_period_calculate"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="20dp"
            android:fontFamily="@font/font"
```

```
    android:textSize="20dp"/>

<Button
    android:id="@+id/btn_help_probationperiod_ok"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:background="@drawable/totalbuttonshape"
    android:text="확인"
    android:textColor="#ffffffff"
    android:textSize="20dp" />
</LinearLayout>
</ScrollView>
```

single_cell_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    android:id="@+id/singleCell"
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="70dp"

    android:orientation="vertical"
    android:padding="8dp"
    >

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/Calendar_day"
        android:text="1"
        android:textSize="20dp"
        android:fontFamily="@font/font"
        android:layout_gravity="center"
        android:textStyle="bold"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:fontFamily="@font/font"
        android:textSize="9dp"
        android:id="@+id/events_id"
        android:textColor="@color/peach"
        android:layout_gravity="center"/>
</LinearLayout>
```