

진도보고서

팀원 20160402 최성원
20160369 박준홍
20212897 도호준

작성일

2022.12.05

성원



UI를 전체적으로 보완함. 아이콘 배치와 이미지, 그리고 레이아웃을 수정하였고 액티비티간 전환이 매끄럽도록 intent Flag 설정과 애니메이션 방식을 추가 하였음.

Fragment간의 전환이 매끄럽지 않은 부분이 있어 이를 대폭 수정함. 기존의 방식으로 하나의 MainActivity로부터 fragment를 참조해 달력의 월간, 주간, 일일 뷰를 나타내는 방식이었는데, item을 선택해 clockListener를 설정하는 과정에서 클릭 이벤트를 받지 못하는 문제가 발생함, 따라서 각각의 뷰를 액티비티로 변경하고 공통적으로 사용하는 Header(화면 상단의 알림, 로고, 설정버튼)을 각각의 액티비티에서 참조하도록 하였음.

일정을 선택하면 다이얼로그가 나오게 했는데(박준홍 담당), 삭제버튼을 클릭하면 해당 일정이 삭제되고, 수정 버튼을 누르면 해당 이벤트의 정보를 가지는 일정 추가 화면이 나오며 수정을 완료하면 해당 이벤트가 지워지고 변경한 내용으로 재생성 되도록 함(HTTP PUT Method와 동일한 방식). 일정을 확인만 하기 위해서는 다이얼로그 바깥 영역을 터치하거나 Back 버튼을 누르면 됨.

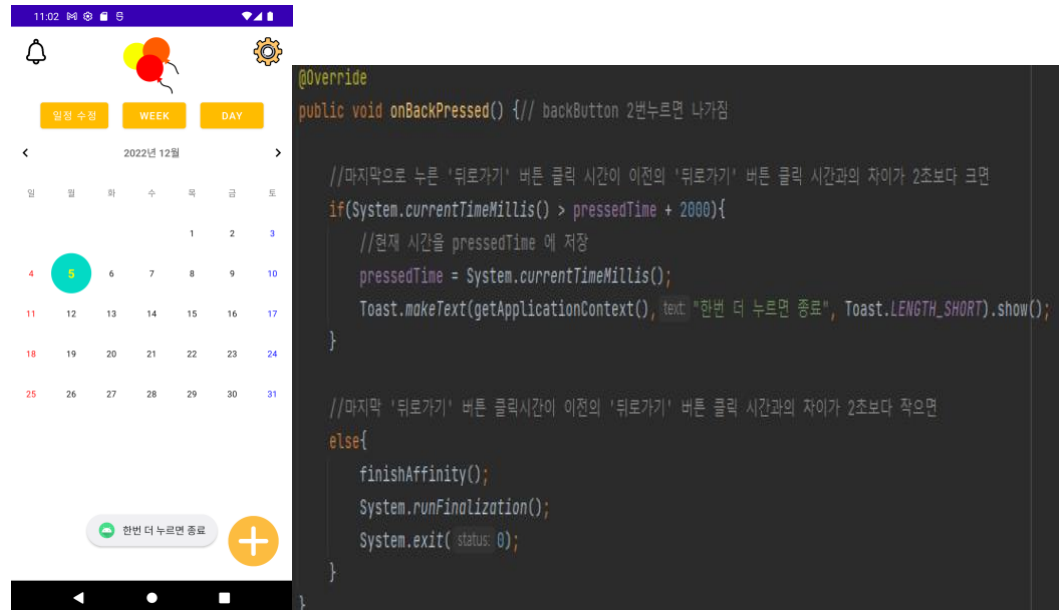
```
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    Log.d(TAG, "onStartCommand: ");
    Thread t1 = new Thread( name: "T1"){
        public synchronized void run(){
            while(true) {...}
        }
    };
    Thread t2 = new Thread( name: "T2"){
        public synchronized void run(){
            while(true) {...}
        }
    };
    t1.start();
    t2.start();
    return START_REDELIVER_INTENT;
}
```

위는 BackgroundService 클래스의 소스코드 일부. 알림을 생성할 이벤트를 찾고 현재 시간과 비교하기 위해 두 개의 작업 스레드를 생성했고 t1에서는 알림을 올릴 가장 최근 이벤트를 감지해 멤버변수에 저장, t2에서는 t1의 저장값이 존재하면 현재 시간과 비교해 해당 시간에 notification을 전달해주는 역할을 함. onStartCommand의 반환값이 START_STICKY이어도 데이터를 가져오는 작업이 실시간으로 동작하지만 그전에 미리 시스템에서 저장된 정보를 가져올 수 있도록 하기 위해 START_REDELIVER_INTENT를 사용



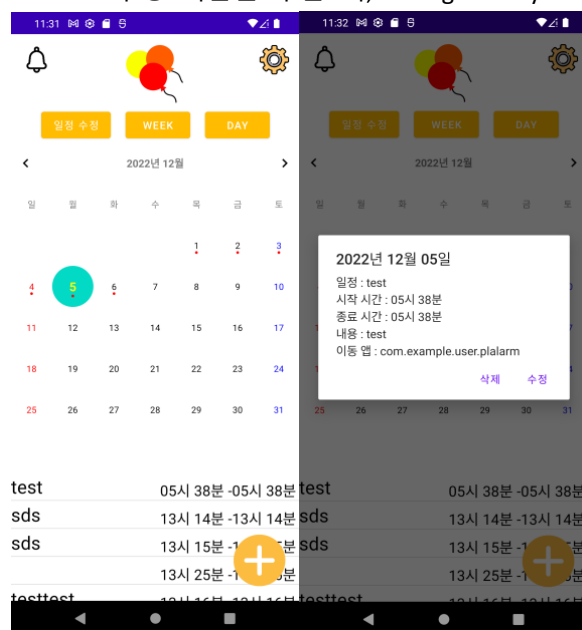
준홍

Main화면에서 뒤로가기 누를 시, Logo_Activity로 넘어가는 것을 한번 누르면 Toast메세지를 띄운 후 한번 더 누르면 어플리케이션이 종료되게 구현.



일정 추가 작업으로 진행 한 것에 대하여 Calendar에 표시

- 오늘 날짜에 대해서 초록색 이미지를 이용하여 표현.
- Firestore에 일정 추가한 것에 대한 data 날짜를 비교 후, 일정이 존재하면 Decorator를 이용하여 빨간 점을 생성
- Calendar Activity에 날짜를 클릭 시, List와 ListAdapter를 이용하여 등록된 일정의 날짜를 하단 부분에 출력
- 조회 하려는 리스트를 클릭 시, AlertDialog로 Firestore의 데이터를 가져와 화면 표현
- 삭제 버튼 누를 시, 각각의 필요한 데이터들과 EventDeletActivity로 화면 Intent. 수정 버튼을 누를 시, Setting Activity로 화면이 전환.



호준

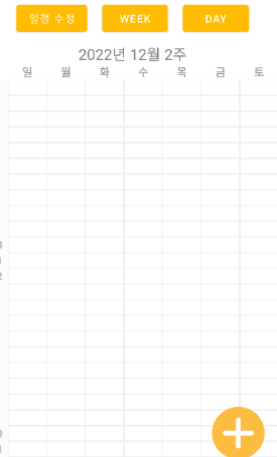
SharedPreferences를 활용하여 상단의 Mute 버튼의 상태와 먼저 볼 뷰의 상태를 저장함, 해당 정보들이 정상적으로

SharedPreferences 파일에 저장되어 있는지 안드로이드 가상 기계의 파일을 탐색하여 확인함.

```
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
  <int name="calendarSpinner" value="1" />
  <boolean name="NOTIFICATION_MUTE" value="true" />
</map>
```



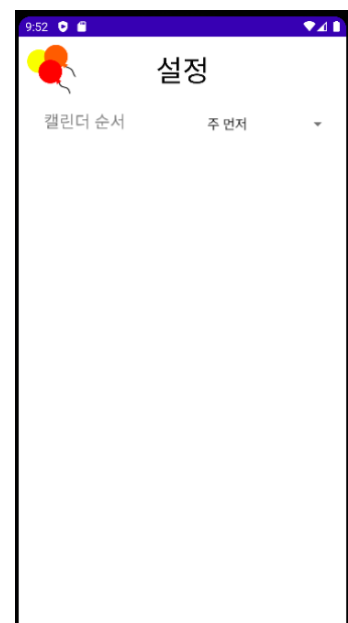
상기된 듯이, Fragment 간의 전환이 매끄럽지 않아 Main Activity의 구성을 바꿈. 월간, 주간, 일간의 3가지 뷰를 전부 Fragment에서 Activity로 바꾸고, 원래 Activity의 상단의 2개의 버튼과 로고 부분을 Fragment화 하여 구현함. 그리고 주간 뷰의 주차를 구하는 메소드의 Logic을 수정하여 정상적인 주차가 나오도록 함.



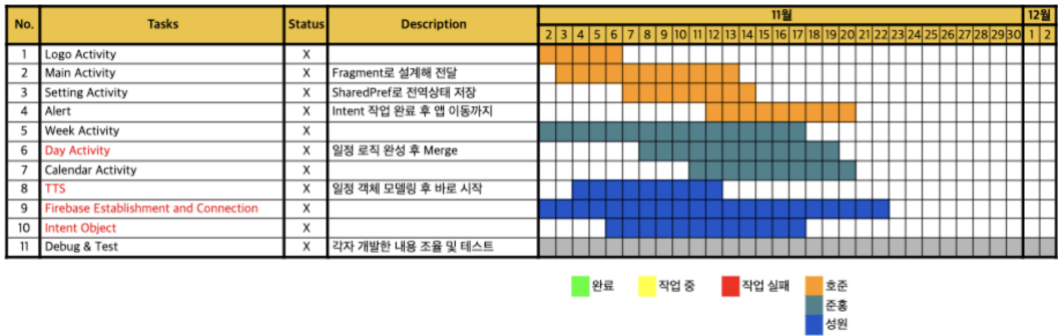
```
String thisTime= todoEvent.getStartTime().substring(0, 16);
SharedPreferences sharedPref = PreferenceManager.getDefaultSharedPreferences(context: BackgroundService.this);
boolean NOTIFICATION_MUTE_BACKGROUND = sharedPref.getBoolean(s: "NOTIFICATION_MUTE", b: false);
if (currentTime.equals(thisTime) && !NOTIFICATION_MUTE_BACKGROUND){
    Intent intent1 = new Intent(packageContext: BackgroundService.this, NotificationActivity.class);
    intent1.putExtra("name", "todoEvent");
    intent1.putExtra("start", thisTime);
    intent1.putExtra("end", todoEvent.getEndTime().substring(0, 16));
    context.startActivity(intent1);
}
```

또한 백그라운드에서 실행하는 Thread에서 SharedPreferences의 값을 참조하여, 만약 Main Activity 상단의 Fragment의 버튼이 눌러 Mute가 활성화 되어있는 경우 Notification이 출력되지 않도록 수정함.

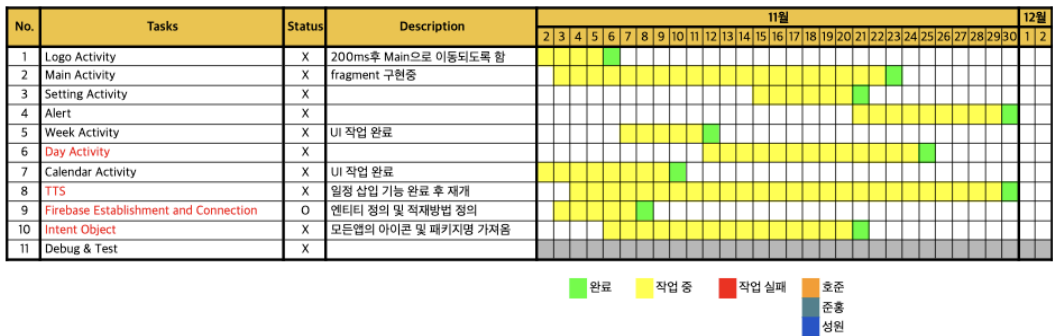
설정에서 먼저 볼 View를 정할 수 있는 선택 기능이 있었는데, 이전까지는 이 기능을 정확히 구현한 바가 없음. 따라서 Logo Activity에서 해당 부분을 SharedPreferences에서 참조하여 선택한 View가 앱을 켜자마자 처음으로 제공되는 View가 되도록 구현함.



프로젝트 전 계획한 일정



현재 개발 일정



Github Project Kanban Board

