**다이나믹케어 키오스크 어플리케이션 가이드**

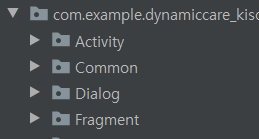
본 어플리케이션은 ㈜다이나믹케어의 컨트롤러와 상호작용하는 키오스크를 조작하기 위하여 개발된 어플리케이션입니다.

어플리케이션 동작을 위한 코드와는 별개로 실제 컨트롤러와 상호 작용하는 메소드들은

도운테크 측에서 제공한 프로토콜을 기반으로 구현되었습니다.

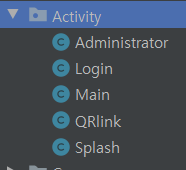
추후 프로토콜의 변경이나 의문점이 있다면 해당 프로토콜 문서를 참고하시면 됩니다.

**패키지 구성**



본 프로젝트는 용도에 따라 크게 4가지의 패키지로 구분되어 있습니다.

**Activity 패키지**



전체적인 UI를 구성하고 있는 액티비티를 포함하고 있는 패키지입니다.

Main, Administrator 액티비티는 프래그먼트를 포함하며, 상단의 액션 바와 하단의 버튼 바를 통해 프래그먼트를 전환시킬 수 있습니다.

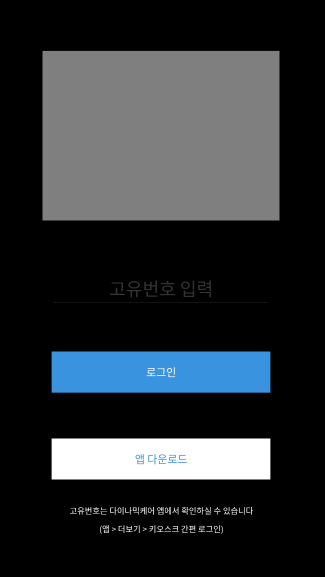
**Splash 액티비티**



리소스를 로딩하는 동안 보여질 화면을 처리하는 액티비티입니다.

해당 화면이 보여질 동안 사운드 파일을 dcSoundPlayer에서 로딩합니다.

**Login 액티비티**



로그인 화면을 처리하는 액티비티입니다.

상단부에 로고 회전 영상을 처리하는 비디오뷰가 있으며, 비디오뷰를 롱 클릭하면 관리자 페이지로 전환합니다.

[로그인] 버튼을 누르면 서버에 고유번호를 전송하여 사용자가 있는지 조회해오고,사용자가 있으면 정보를 가져오고 화면을 Main 액티비티로 전환합니다.

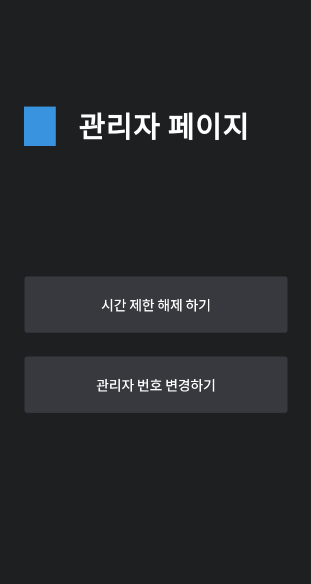
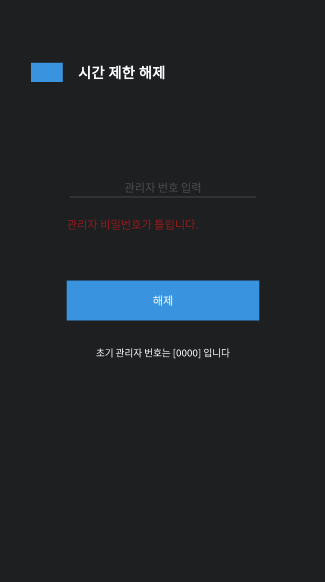
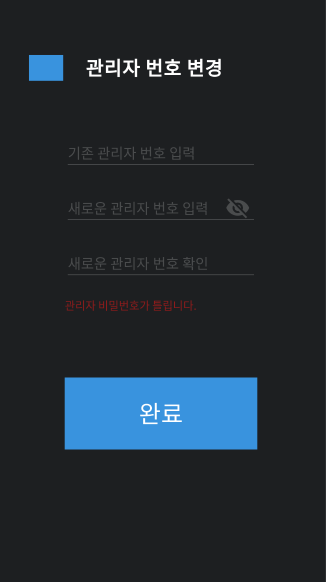
**QRlink 액티비티**



Login 액티비티에서 [다운로드] 버튼을 누르면 전환되는 액티비티입니다.

사용자가 앱을 다운로드 받을 수 있는 링크를 제공합니다.

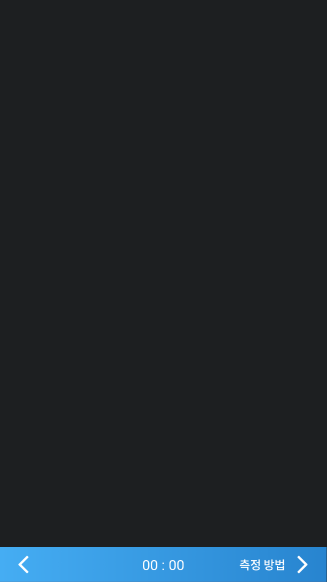
**Administrator 액티비티**

위와 같은 세 개의 프래그먼트 표시를 담당하는 액티비티입니다.

Login 액티비티의 로고를 롱클릭 했을 때에 나타나는 페이지이며, 관리자 권한이 필요한 여러 작업들을 수행하기 위한 화면입니다.

**Main 액티비티**



본 어플리케이션의 중심이 되는 액티비티입니다. 운동 기능과 측정 기능 모두가 위 액티비티에서 이루어집니다. 다수의 프래그먼트를 전환하는 기능을 포함하고 있으며,하단 바를 통해 프래그먼트를 전환할 수 있습니다. 중앙에는 타이머가 위치하여 사용자의 현재 남은 사용시간을 표시하는 역할을 합니다.

추가적으로 담당하는 역할로는

-프래그먼트에서 발생하는 사운드를 재생하는 역할

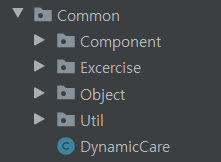
-USB 연결 및 해제를 인식하고 상호작용하는 역할

-컨트롤러로부터 전송된 ACK를 인식하고 처리한 뒤 현재 보여지는 프래그먼트로 보내는 역할

-현재 선택된 운동 형태 및 운동 종목을 저장하는 역할

등이 있습니다.

**Common 패키지**



전체 액티비티 및 프래그먼트에서 광범위하게 사용되는 공통적인 객체 또는 메소드들을 정리해놓은 패키지입니다.

**DynamicCare 클래스**

Application 클래스를 상속받은 클래스로서, 전체 액티비티나 프래그먼트에서 접근이 필요한 정보를 담고 있습니다.

포함하는 주요 정보로는

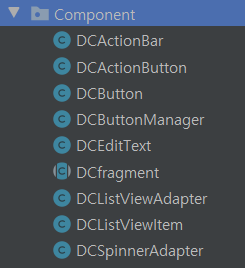
-사용자 정보

-제한 시간

-관리자 비밀번호

등이 있습니다.

**Common.Component 패키지**



여러 프래그먼트에서 공통적으로 사용되는 커스터마이징 뷰 또는 부모 클래스 등을 포함하는 패키지입니다.

**-DCActionBar**



Main 액티비티 상단에 위치하는 액션바를 구현하는 클래스입니다.

각 프래그먼트의 제목을 중앙에 표시하며,

홈 버튼,시간 연장 버튼,로그아웃 버튼의 클릭 이벤트를 처리합니다.

-**DCButton**



주요 프래그먼트에서 사용되는 버튼을 구현하는 클래스입니다.

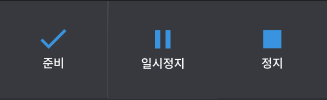
일반적인 버튼을 포함하고,해당 버튼이 눌려서 토글되었을 때의 표시될 버튼 이미지,

해당 버튼을 눌렸을 때 표시할 body 이미지를 저장합니다.

눌린 버튼을 참조하는 static 객체를 가지고 있어, 특정 그룹 내에서 하나의 버튼만 눌려야 하는 상황에 사용됩니다.

특정 화면의 ImageView와 상호작용하여 버튼을 눌렀을 때 이미지를 변경하도록 처리합니다.

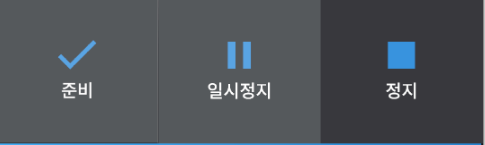
**-DCActionButton**



DCButton과 마찬가지로 눌렸을 때의 이미지를 저장하고 있는 버튼입니다.

하지만 DCButton과 같이 하나의 버튼만 눌릴 필요는 없는 위와 같은 버튼들에 사용됩니다.

**-DCButtonManager**



기능 수행 상태에 따른 버튼들의 활성화/비활성화 여부를 결정하는 클래스입니다.

해당 클래스에 등록된 버튼들은 상태 전환에 따른 제어에 영향을 받게 됩니다.

-수행 상태

*Clear*, *StartSetting*, *Setted*, *Ready*, *Excercise*, *onRest*, *Paused*, *Stop*

Clear:어떤 버튼도 선택되지 않은 상태로,준비/시작/정지와 같은 조작버튼은 비활성화 처리합니다.

StartSetting:특정 종목이 선택된 상태로,정지 외의 버튼들을 비활성화 처리함으로서 컨트롤러의 바 또는 와이어가 바른 자리에 위치할 때까지 다른 버튼으로의 접근을 차단합니다.

Setted:바 또는 와이어가 자리에 위치한 경우로서,비활성화 시켰던 다른 버튼들을 활성화 처리합니다.

Ready:준비 버튼을 누른 상태입니다.

Exercise:운동 수행 중인 상태입니다.일시정지 버튼이 활성화됩니다.

onRest:휴식 중인 상태입니다.운동 모드의 경우 버튼 대신 몇 초 남았는지를 표시하는 텍스트로 전환합니다.

Paused:일시 정지된 상태입니다.

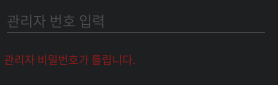
Stop:정지 버튼을 눌렀을 때의 동작을 수행합니다.

**-DCEditText**



어플리케이션에서 사용되는 EditText입니다.

글자 입력을 하면 밑줄을 파란색으로 변경하는 역할을 합니다.



생성자에 TextView를 추가적으로 등록할 경우에

글자 입력 이벤트에 따라 해당 텍스트뷰 표시 여부를 결정짓는 역할도 포함합니다.

**-DCListViewAdapter &DCListViewItem**



예정된 일정이나 플랜을 불러올 때에 쓰이는 커스텀 리스트뷰용 어댑터 & 아이템입니다.

**-DCSpinnerAdapter**



등속성 운동 모드의 레벨을 표시하기 위한 커스텀 스피너 어댑터입니다.

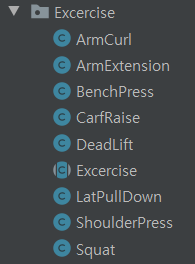
**-DCfragment**

모든 프래그먼트가 공통으로 포함하고 있어야 할 메소드나 객체들을 정의하고 있는 부모 클래스입니다.

Main의 하단 바에서 back 버튼 또는 Next 버튼을 눌렀을 때 전환될 프래그먼트를 정의하거나,

다른 프래그먼트들이 필수로 구현해야할 추상 메소드들을 정의합니다.

**Common.Excercise 패키지**



선택된 운동 종목에 대한 정보를 제공하는 클래스들을 포함하고 있는 패키지입니다.

**Exercise 클래스**

운동 종목들이 필수적으로 제공해야 할 메소드들을 정의하고 있는 추상클래스입니다.

제공되는 정보로는

-측정방법에 대한 설명문

-운동 이름

-운동으로 발달시킬 수 있는 근육부위 이름

-해당 근육부위를 표현하는 인체 이미지

-튜토리얼 비디오의 영상 uri

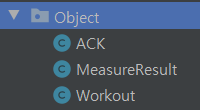
-컨트롤러에서 사용되는 운동모드 번호

-실제 DB 상에 저장되는 코드

가 있습니다.

본 클래스 이외의 클래스들은 Exercise 클래스를 상속하고 추상 메소드를 구현합니다.

**Common.Object 패키지**

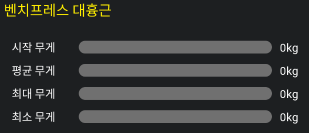


각 프래그먼트 및 컨트롤러와 상호작용하기 위한 객체들을 포함하고 있는 패키지입니다.

**ACK 클래스**

컨트롤러에서 키오스크로 전송되는 신호를 파싱한 데이터를 항목별로 분류해서 저장해놓은 객체입니다.

**MeasureResult 클래스**



측정 된 데이터를 입력받아 평균 무게/최대 무게/최소 무게/시작 무게를 산출하는 객체입니다.

위 화면에서 보여지는 데이터를 저장합니다.

**Workout 클래스**

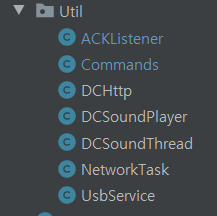


수행할 운동/수행된 운동에 대한 정보를 담는 객체입니다.

서버로부터 수신받은 플랜/일정 등을 저장하여 리스트 뷰 항목으로 사용되기도 하고,

운동이 끝난 뒤에 서버로 운동 기록을 송신하는 데에 이용되기도 합니다.

**Common.Util 패키지**



여러 프래그먼트 및 액티비티에서 범용적으로 특정 절차를 처리하기 위해 사용되는 클래스들을 모아놓은 패키지입니다.

**ACKListener 클래스**

컨트롤러에서 수신된 ACK를 입력받는 핸들러입니다.

static으로 정의된 ParseACK 메소드를 포함하고 있으며, 수신받은 신호를 파싱하여 신호 종류에 따라 각각의 데이터를 분류하고 저장해 ACK 객체로 반환합니다.

**Commands 클래스**

키오스크에서 컨트롤러로 내릴 수 있는 명령어에 대하여 정의하는 클래스입니다.

컨트롤러의 동작을 필요로 하는 기능이 있을 경우에 사용됩니다.

도운테크 측에서 제공된 프로토콜을 기반으로 하고 있으며,

명령어의 Checksum을 구하는 메소드를 포함하고 있습니다.

명령어들은 static으로 선언되어 객체 선언없이 사용될 수 있습니다.

**DCHttp 클래스**

서버와 통신이 필요한 요청을 받는 객체입니다.

요청에 걸맞는 url과 파라미터를 입력하여 NetworkTask를 실행합니다.

**NetworkTask 클래스**

서버와 직접적으로 통신하는 AsyncTask 객체입니다.

POST 방식으로 통신하며, 파라미터로 들어온 url로 접속하여 데이터를 전송해주는 역할을 합니다.

**DCSoundPlayer 클래스**

키오스크 내의 컴포넌트를 누를 때 나오는 효과음 및 안내 음성을 재생할 때 사용되는 객체입니다.

[resources]-[raw] 폴더 내의 사운드 파일을 로딩하고 몇 초 동안 재생할 것인지를 정의합니다.

**DCSoundThread 클래스**

안내 음성과 같이 특정 사운드를 순차적으로 재생할 필요가 있을 때 사용되는 객체입니다.

Resource id의 배열을 입력으로 받아 해당 배열을 순차적으로 재생합니다.

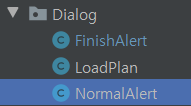
**UsbService 클래스**

안드로이드 운영체제에서 USB입력을 통한 통신을 지원하기 위한 객체입니다.

<https://github.com/felHR85/UsbSerial> 소스를 응용하여 만들어졌으며,

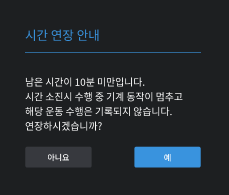
명령어를 전송할 시에 USB 연결 여부와 인터넷 연결 여부를 검사하는 기능을 포함하고 있습니다.

**Dialog 패키지**



액티비티 및 프래그먼트와 별개로 화면에 띄워지는 경고창을 표시하는 클래스들을 모아놓은 패키지입니다.포함된 각 클래스는 Dialog를 확장하였습니다.

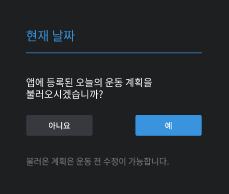
**FinishAlert 클래스**



시간 제한이 끝나 사용을 종료하기 전 사용자에게 시간 연장 의사를 묻는 안내창입니다.

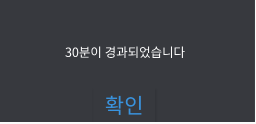
예를 누를 경우 시간 연장 화면으로 전환합니다.

**LoadPlan 클래스**



운동 모드로 진입하기 전 예정된 플랜이나 일정이 있을 경우 사용자에게 알려주고, 불러올 것인지를 묻는 경고창입니다.

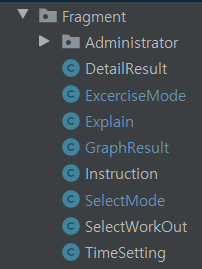
**NoarmalAlert 클래스**



확인 버튼 하나만 필요한 경고창을 띄울 때에 주로 사용되며, 위의 텍스트를 용도에 맞게 바꾸어 띄울 수 있는 경고창입니다.

시간 만료, 인터넷 연결 접속 실패, USB 연결 실패 경고창에 사용됩니다.

**Fragment 패키지**

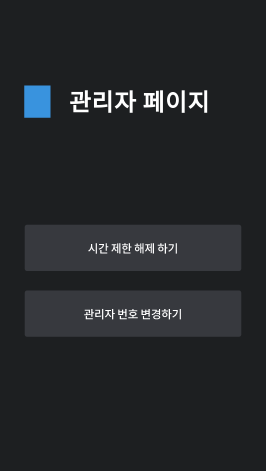


Main 또는 Administrator 액티비티에 포함되는 프래그먼트들을 포함하고 있는 패키지입니다.

**Fragment.Administrator 패키지**

Administraor에 포함되는 관리자 모드 프래그먼트들을 포함하는 패키지입니다.

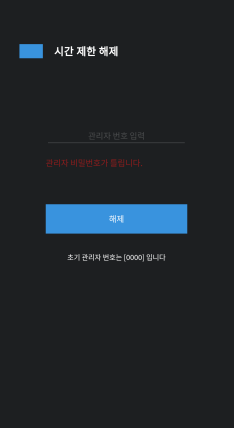
**Authentification 클래스**



관리자 페이지의 초기 화면을 표시하는 프래그먼트입니다.

버튼을 누르면 해당 화면으로 전환됩니다

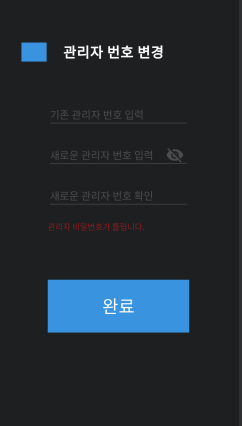
**LimitOff 클래스**



시간 제한을 해제하는 기능을 수행하는 클래스입니다.

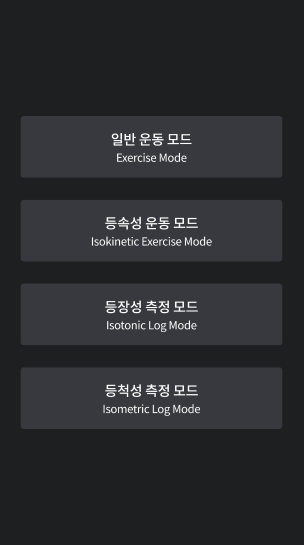
사전에 등록된 관리자 번호와 대조하여 일치할 경우 제한을 해제합니다.

**ModifyAdmin 클래스**



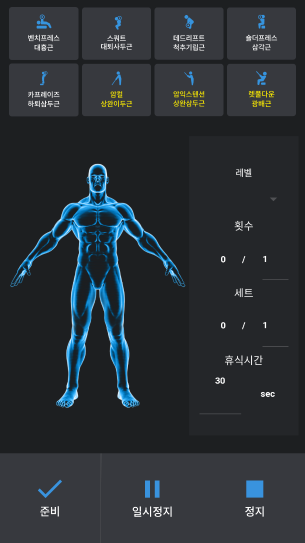
관리자용 비밀번호를 변경하는 프래그먼트입니다.

**SelectMode 클래스**



각 모드에 대한 선택지를 제공하는 프래그먼트입니다. 어떤 모드를 선택한 경우 컨트롤러로 해당 모드에 대한 명령이 전송됩니다.

**ExerciseMode 클래스**

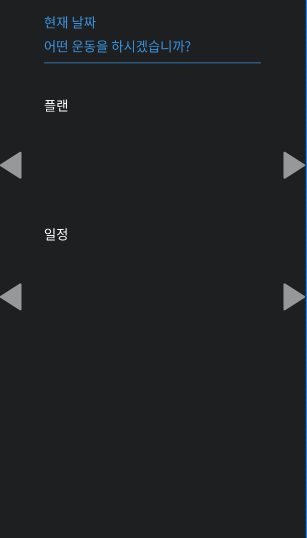


운동 모드의 기능 전체를 수행하는 프래그먼트입니다.

main에서 전송된 ack를 자체 해석하는 메소드를 포함하며,

조작 버튼을 눌렀을 때에 컨트롤러로 명령을 보내고 수신된 ACK에 따라 운동을 진행하는 역할을 합니다.

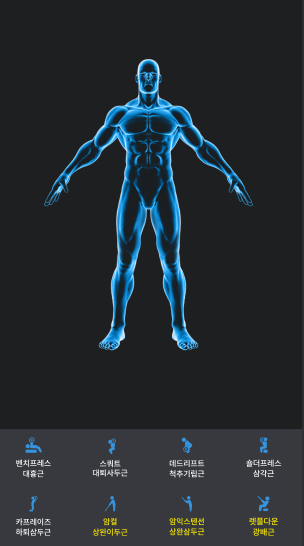
**SelectWorkout 클래스**



예정된 일정이나 플랜을 불러왔을 때 보여지는 프래그먼트입니다.

서버로부터 받은 플랜과 일정에 대한 정보를 리스트뷰에 매핑하는 역할을 합니다.

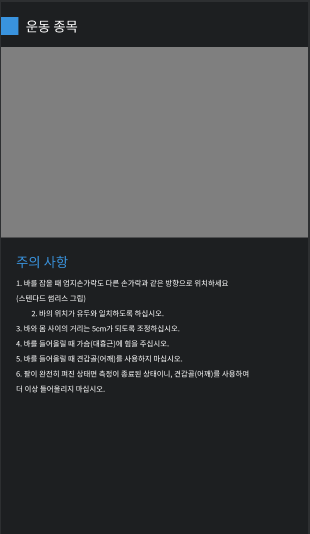
**Explain 클래스**



측정 모드의 운동 종목을 선택할 수 있는 프래그먼트입니다.

버튼을 누르면 그 운동에 해당하는 부위를 설명하는 안내문을 표시합니다.

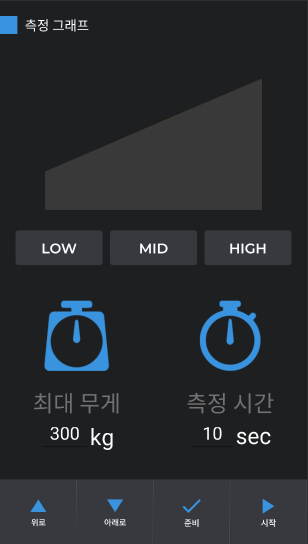
**Instruction 클래스**



선택한 종목에 해당하는 튜토리얼 영상과 안내문을 표시하는 프래그먼트입니다.

상단에는 튜토리얼 영상을 재생하는 VideoView가 위치합니다.

**GraphResult 클래스**



측정을 수행하는 프래그먼트입니다

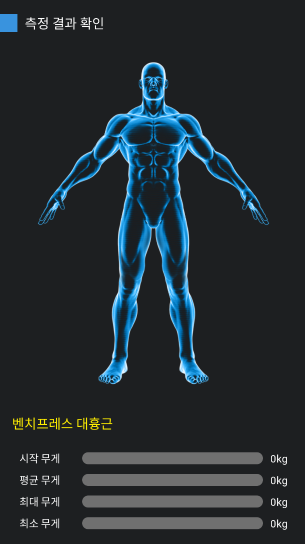
각 버튼을 누르면 컨트롤러에도 명령이 전송됩니다.

최대 무게와 측정 시간을 설정하면 DynamicCare 클래스에 그 값이 저장되어 보존되며,

측정이 시작한 뒤에 컨트롤러로부터 들어오는 신호를 파싱하여 최대값을 ProgressBar에 업데이트합니다.

측정이 종료되면 결과를 서버로 전송합니다.

**DetailResult 클래스**

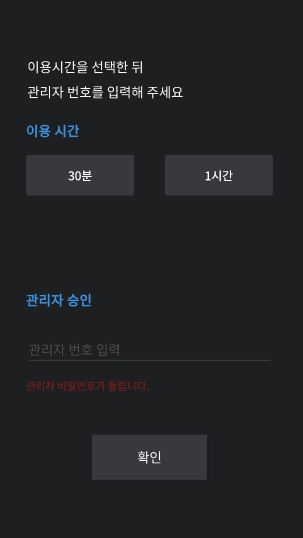


측정 결과를 사용자에게 보여주는 프래그먼트입니다.

GraphResult 프래그먼트로부터 받은 MeasureResult 객체를 사용하여,

Thread를 통해 애니메이션 효과를 구현합니다.

**TimeSetting 클래스**



시간 연장 기능을 수행하기 위한 프래그먼트입니다.

첫 로그인 시에 가장 먼저 표시되는 프래그먼트이며, 이를 통과하지 않고는 다른 프래그먼트에 진입될 수 없습니다.

ActionBar에 있는 시간 연장 버튼을 통해서도 진입할 수 있으며, 이 경우 이전 프래그먼트를 보존하여 다시 되돌아갈 수 있도록 합니다.

확인 버튼을 누를 경우 선택된 시간이 하단 바의 사용 시간에 추가됩니다.