

이상운동증 진단 어플리케이션

개발자 메뉴얼

황보효정, 윤다은, 하재경, 배수한, 권효정, 장지현, 권다은, 안민규

한동대학교

ahnbcilab@gmail.com

Github : https://github.com/YunPC/TremorQuantification_ver2-App-master.git

목차

1. 개발 환경 구성
2. 개발 개요
3. 앱 디자인 설계
4. System 구조
5. 앱 전체 구조
6. 앱 상세 설명
7. 데이터 베이스 구조
8. 데이터 베이스 상세 설명
9. 정량화 방법 설명

1 개발 환경 구성

1.1 개발 하드웨어 정보

1.1.1 컴퓨터 버전 : Windows 10 Pro

1.1.2 시스템

1.1.2.1 프로세서 : Intel® Core(TM) i7-7700 CPU @ 3.60GHz 3.60 GHz

1.1.2.2 RAM : 16.0GB

1.1.2.3 시스템 종류 : 64bit 운영 체제 x64 기반 프로세서

1.2 개발 디바이스

1.2.1 정보 : Galaxy Tab A SM-T510

1.2.2 해상도 : 1920x1200

1.3 개발 프로그램

1.3.1 정보 : Android Studio

1.3.2 버전 : 3.1.2

1.3.3 SDK 버전 : CompileSdkVersion : 28, minSdkVersion : 26, targetSdkVersion : 27,
buildToolsVersion : '28.0.3'

1.4 데이터 베이스 프로그램

1.4.1 정보 : Fire Base

1.4.2 버전

'com.google.firebase:firebase-storage:16.0.4'

'com.google.firebase:firebase-core:16.0.6'

'com.google.firebase:firebase-auth:16.1.0'

'com.google.firebase:firebase-firestore:18.0.0'

'com.google.firebase:firebase-database:16.0.6'

2 개발 개요

전 세계적으로 급증하고있는 고령화에 따라 노인질환중 이상운동증 (파킨슨병 , 본태진전증)이 늘어나고 있는 추세이다. 그리하여 이상운동증을 진단받고 치료받기위해 의료진을 찾아간 환자들에게 더 나은 치료를 받게 하는 것과 의료진에게 오진의 확률을 줄여주기 위한 어플리케이션을 개발 하고자 한다.

2.1 문제 정의

이상운동증을 진단받고 치료받기 위해서는 pen and paper 형식의 설문조사인 UPDRS와 CRTS가 있다. 이 두개의 설문조사는 환자의 운동성을 관찰하여 점수를 부여하는 방식이다. 하지만 이 방법은 증상의 경중을 나누는 기준이 객관적인 수치가 아닌 전문가의 주관적인 판단에 의존하기 때문에 같은 환자를 진단하더라도 임상이에 따라 다른 진단을 내릴 수 있다.

2.2 개발 목표

- 1) 파킨슨병 환자 진단 시 수행되는 여러 임상적기준들을 공학적 척도를 기반으로 정량화(수치화)한다. 이를 통해 기존에 의사의 주관적인 판단에 의지했던 진단 방법을 최소화하며, 범용적인 기준을 세운다.
- 2) 환자를 진단 시 어플리케이션을 활용하여 의사들에게 편의성을 제공한다. 종이와 펜을 이용한 아날로그식 방식을 탈피하고 디지털화된 어플리케이션을 통해 환자 개개인의 지속적인 모니터링을 돕는다. 본 어플리케이션은 공학적 지식이 없는 사용자도 그래픽을 통해 직관적으로 알게 하는데 목적이 있다.
- 3) 축적된 데이터는 파킨슨병의 조기진단을 위한 연구 데이터로 활용한다. 데이터 베이스 기반의 어플리케이션을 구축하여 정량화된 데이터를 체계적으로 관리하며 이는 후에 파킨슨병 연구를 위한 데이터로 활용한다.

2.3 요구 사항 및 제약 조건

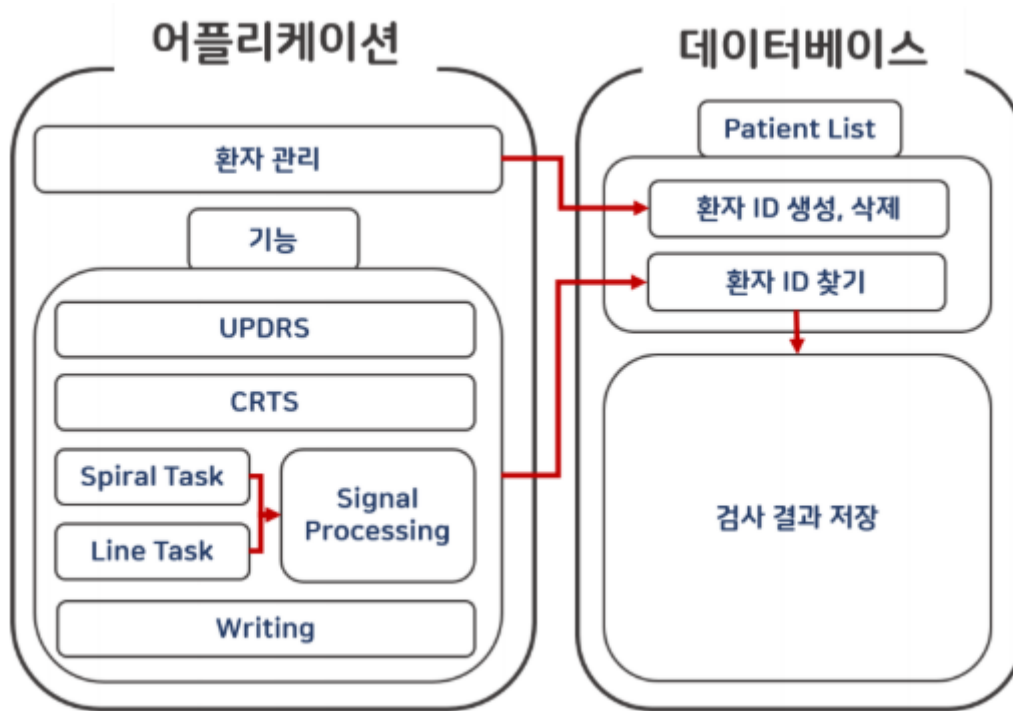
본 프로젝트는 이상운동증 진단을 정량화 하는 알고리즘을 연구하고, 모니터링이 가능한 데이터베이스 기반의 어플리케이션으로 구현하는 것이다.

제약조건으로는 본 어플리케이션의 알고리즘을 검증하는 차원에서 환자를 모사한 데이터 , 즉 환자의 데이터가 아닌 연구자가 만든 데이터이므로 더 확실한 검증을 하기위해 차후 실제 환자의 데이터를 받아서 검증할 예정이다.

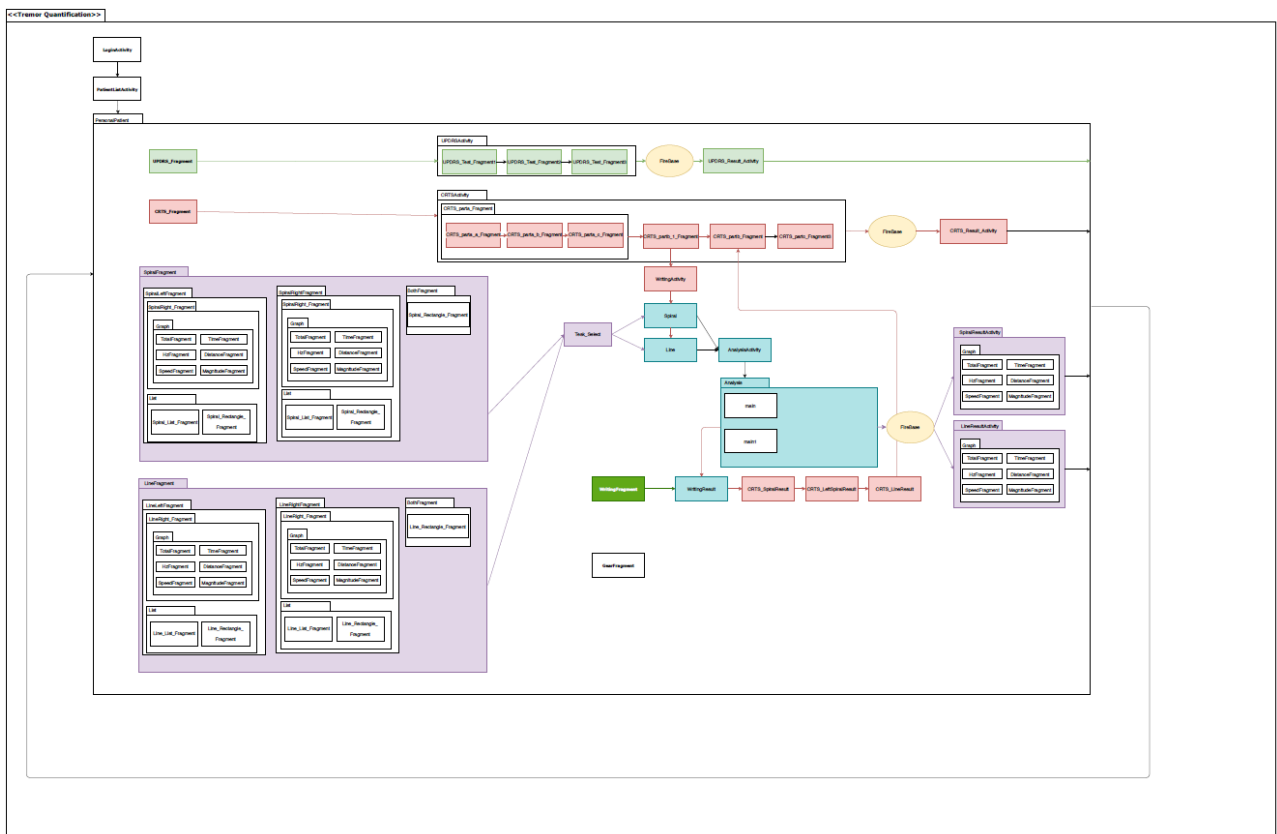
3 SW 디자인 설계

<https://www.figma.com/file/w6dggw8rcvTfuvTote5Z9L/%EC%9D%B4%EC%83%81%EC%9A%B4%EB%8F%99%EC%A6%9D-App-2%EC%B0%A8?node-id=0%3A1>

4 System 구조



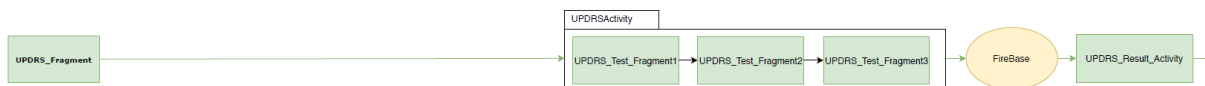
5 앱 전체 구조



LoginActivity : 처음 시작하는 화면. 구글 계정을 이용하여 로그인 및 계정 변경. 로그인 후 firebase 데이터베이스에 연동되어 계정 마다 환자 관리 가능. 로그인을 한 번 한 경우, 앱을 켜를 때 자동 로그인이 됨. firebase에서 제공하는 로그인, 계정 변경 코드 사용.

PatientListActivity : 환자 리스트가 나오는 화면. 환자 등록, 삭제, 검색, 계정 변경, 로그아웃 가능. 이때 로그인한 구글 계정의 key를 가지고 구글 계정마다 따로 환자를 저장하고, 관리할 수 있도록 함. RecyclerView를 사용하여 리스트 제시. 리스트를 클릭하면 클릭한 리스트의 환자 페이지로 이동. 이때 환자 ID와 이름을 넘겨줌.

PersonalPatient : 환자 상세 정보 화면. 검사 종류 마다 다른 tab 화면을 가지기 위해 fragment 사용. 7개의 fragment 존재. 해당 검사에서 검사를 한 번도 하지 않았을 시 NonTaskFragment로 화면 전환, 해당 tab 검사 type을 전달하여 '새로운 검사 진행하기' 버튼을 누를 때 각각 다른 검사 activity로 이동할 수 있도록 함. tab 버튼을 누를 때 마다 firebase가 동작하여 검사를 했는지 확인.

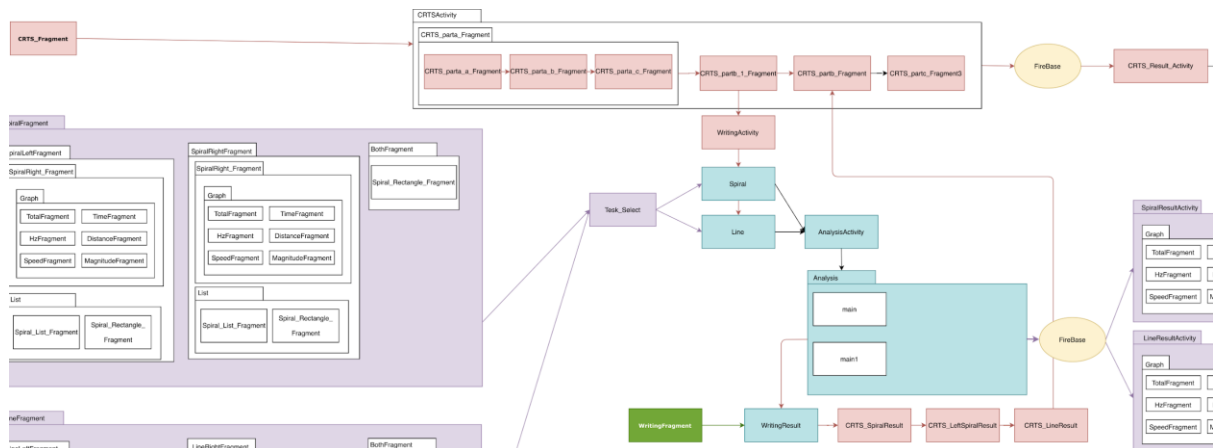


UPDRS_Fragment : 이 fragment에서 updrs검사와 결과를 확인할 수 있음. 마지막 검사 결과와 그래프, 리스트는 firebase에 저장된 값을 가져옴. '새로운 검사 진행하기' 버튼을 누를 시 검사 진행 activity로 이동. 리스트는 역대 검사 기록을 보여줌. 해당 리스트 클릭 시 그 날짜에 검사 결과 항목을 자세히 확인할 수 있음.

UPDRSActivity : UPDRS 검사 화면. 세개의 fragment를 가지고 있어, 세 화면으로 검사가 나눠 이루어짐. 각 다음 Fragment로 갈 때 마다 UPDRSActivity에서 설문 검사 (라디오 버튼) 값을 저장. 마지막 fragment에서 결과 activity로 이동시 UPDRSActivity에서 검사 결과 값을 더하여 firebase에 저장.

UPDRS_Test_Fragment1, UPDRS_Test_Fragment2, UPDRS_Test_Fragment3 : 검사 설문지가 나오는 fragment.

UPDRS_Result_Activity : 검사 결과가 나오는 activity. Firebase에서 데이터를 가져와 총 점수를 나타내는 추이 그래프와 검사 상세 점수를 확인할 수 있음.



CRTSActivity : CRTS 검사 화면. 네개의 fragment를 가지고 있어, 세 화면으로 검사가 나눠 이루어짐. 각 다음 Fragment로 갈 때 마다 CRTSActivity 에서 설문 검사 (라디오 버튼) 값을 저장. 마지막 fragment에서 결과 activity로 이동시 CRTSActivity 에서 검사 결과 값을 더하여 firebase에 저장.

CRTS_parta_Fragment, CRTS_partb_1_Fragment, CRTS_partb_Fragment, CRTS_partc_Fragment : 검사 설문지가 나오는 fragment. CRTS_partb_1_Fragment는 그리기 검사 전 화면. CRTS_partb_Framgnet는 그리기 검사 후, 설문지 형태 화면.

CRTS_parta_a_Fragment, CRTS_parta_b_Fragment, CRTS_parta_c_Fragment : CRTS_parta_Fragment를 세 개의 스와이프가 가능한 fragment로 나눔. 스와이프 할 때 마다 값을 CRTS_parta_Fragment에서 저장.

CRTS_partb_1_Fragment : 그리기 검사 버튼 누를 시 그리기 검사를 진행.

WritingActivity : 글씨 따라 그리는 화면. 따라 쓴 그림은 이미지 저장. (firebase storage 활용)

Spiral : Toast 메시지로 주의 안내문이 이미지로 띄워진다. 나선을 따라 그리는 화면. 이때 그릴 때 마다 좌표 값 저장. 다음 버튼 누르면 이 좌표값을 데이터로 가지고 AnalysisActivity로 이동하여 5가지 measure 로 수치화 및 정량화. 그리고 그린 그림은 캡처되어 파이어베이스로 저장된다.

Line : Toast 메시지로 주의 안내문이 이미지로 띄워진다. 선을 따라 그리는 화면. 이때 그릴 때 마다 좌표 값 저장. 다음 버튼 누르면 이 좌표값을 데이터로 가지고 AnalysisActivity로 이동하여 5가지 measure 로 수치화 및 정량화. 그리고 그린 그림은 캡처되어 파이어베이스로 저장된다.

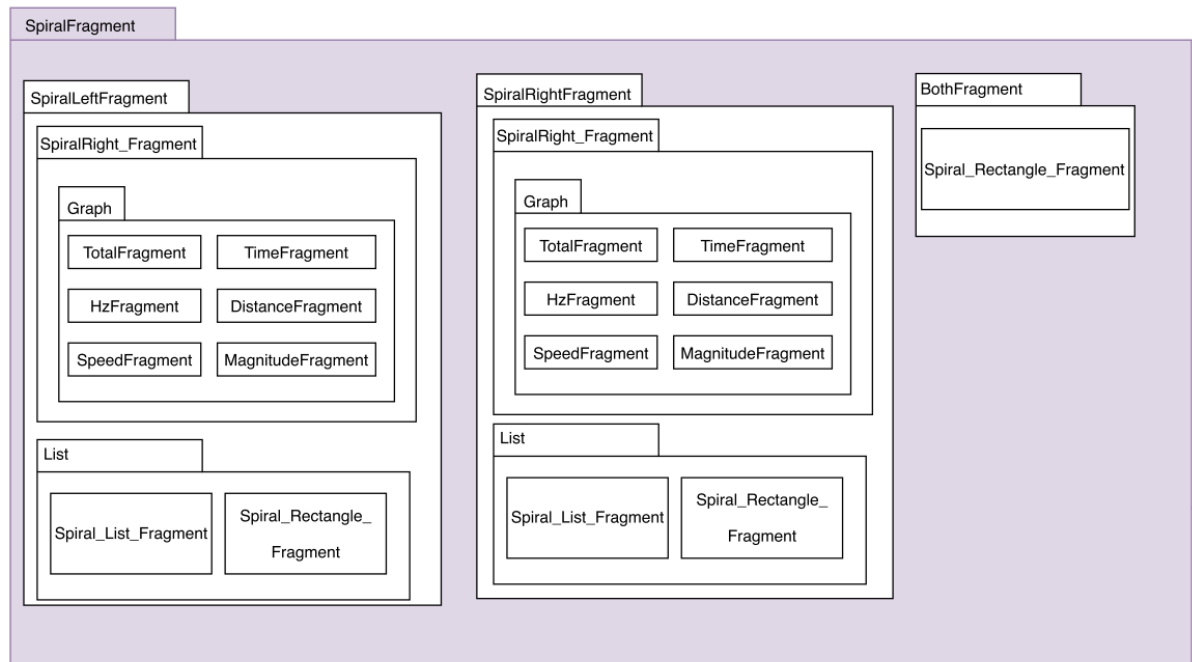
AnalysisActivity : CRTS에서 검사를 진행했는지, Spiral, Line에서 검사를 진행했는지 value를 전달 받아 해당되는 기준으로 검사 결과 측정.

WritingResult : 그린 그림을 보여주는 화면. 이때 spiral, line 검사 결과를 데이터로 전달 받으면서, 다음 acitivity로 전달. 라디오 버튼 체크 필요. 수정 모드인지 구별 하여 수정 모드일 때 체크한 라디오 버튼 값이 체크 되도록 제시.

CRTS_SpiralResult, CRTS_LeftSpiralResult, CRTS_LineResult : 그린 그림을 보여주고, 전달 받은 데이터로 수치 값을 제시. 라디오 버튼 체크 필요. 이때 값들은 모두 firebase에 저장. 수정 모드인지 구별하여 수정 모드일 때 체크한 라디오 버튼 값이 체크되도록 제시.

CRTS_partb_Fragment : 측정 검사를 모두 저장하고, 체크한 라디오 버튼 값이 화면에 반영되도록 함.

CRTS_Result_Activity : 검사 결과가 나오는 activity. Firebase에서 데이터를 가져와 총 점수를 나타내는 추이 그래프와 검사 상세 점수를 확인할 수 있음. 이때 저장된 그림도 제시할 수 있도록 함.



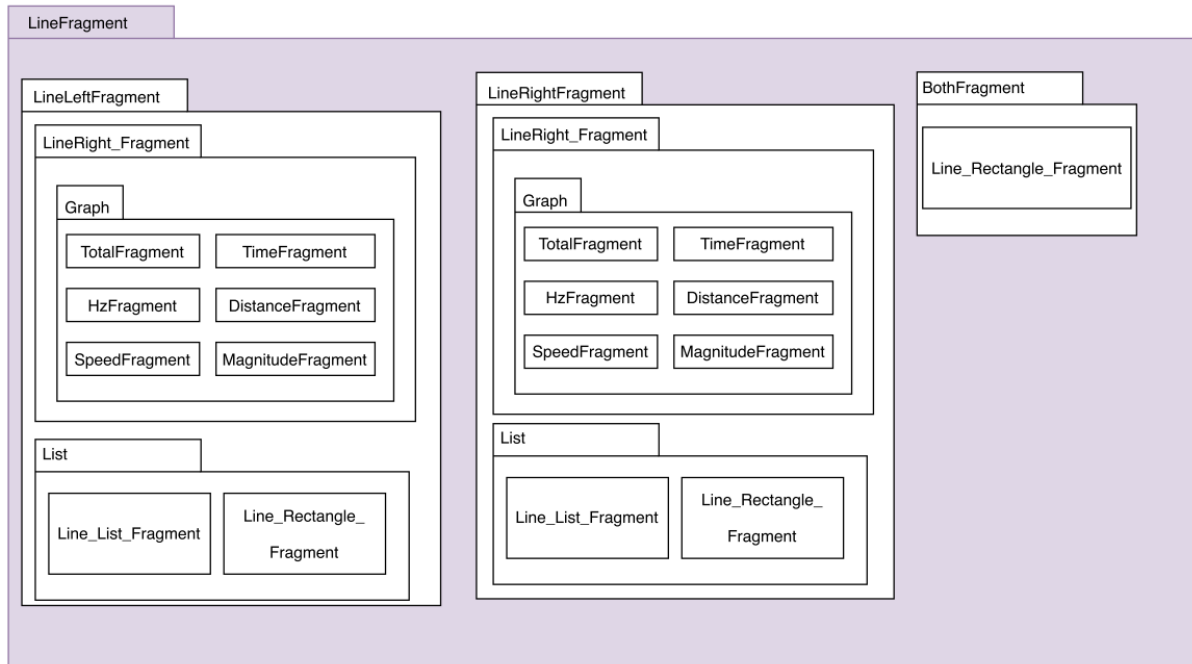
SpiralTask_Fragment : Spiral 검사가 왼손인지, 오른손인지, 양손인지에 따라 내부적으로 Fragment가 나뉜다. 그 안에, measure값을 그래프로 나타내는 Fragment와 List영역의 Fragment로 나뉘어져 있다.

SpiralLeftFragment, SpiralRightFragment, SpiralBothFragment : 왼손, 오른손, 양손 탭으로 나뉘는 Fragment이다. 나선 검사를 시작할 수 있고, 이 SpiralLeftFragment, SpiralRightFragment 안에 이전 결과 내역을 알 수 있는 measure 값을 그래프로 나타내는 Fragment와 List 영역의 Fragment로 나뉜다. SpiralBothFragment에서는 그래프 결과 값없이 그림 리스트로만 확인이 가능하다.

SpiralRight_Fragment, SpiralLeft_Fragment : 이전 검사 결과를 알 수 있는 measure 값을 그래프로 나타내는 Fragment이다. 각 6개의 탭마다 Fragment가 존재하여, 그래프 값을 확인할 수 있다.

TotalFragment, HzFragment, MagnitudeFragment, DistanceFragment, SpeedFragment, TimeFragment : 각 measure 별로 결과 값을 그래프로 알려준다. 파이어베이스에서 값을 실시간으로 받아와서 그래프를 생성한다.

Spiral_List_Fragment, Spiral_Ractangle_Fragment, Spiral_Both_Rectangle_Fragment : 일반 List와 그림 List는 각 다른 Recycler View Adapter를 사용한다. Spiral_Ractangle_Fragment 안에는 왼손, 양손 Recycler View를 따로 사용하여 그림이 나타나도록 한다.



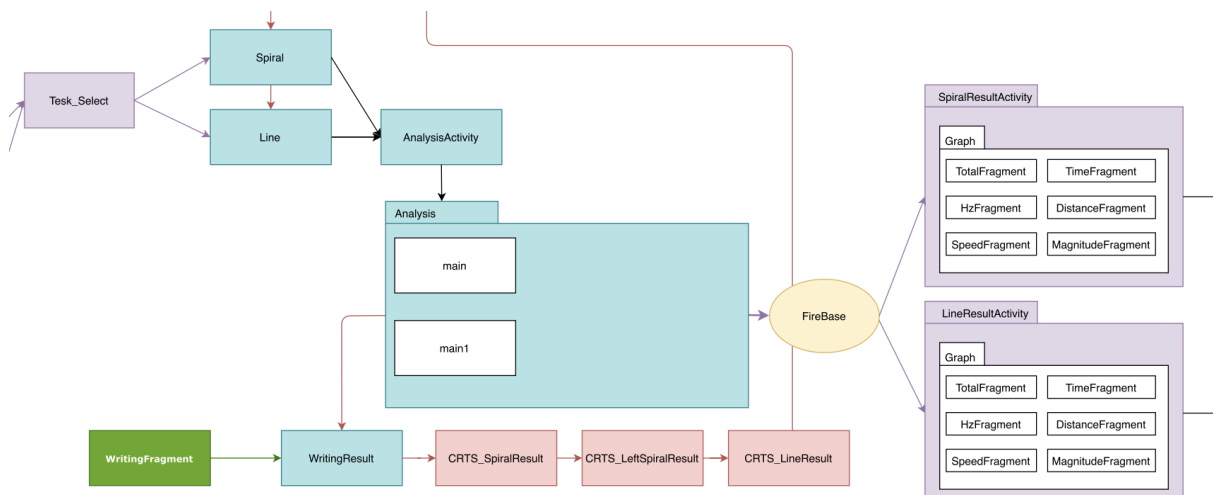
LineTask_Fragment : Line 검사가 왼손인지, 오른손인지, 양손인지에 따라 내부적으로 Fragment가 나뉜다. 그 안에, measure값을 그래프로 나타내는 Fragment와 List영역의 Fragment로 나뉘어져 있다.

LineLeftFragment, LineFragment, LineBothFragment : 왼손, 오른손, 양손 탭으로 나뉘는 Fragment이다. 선 검사를 시작할 수 있고, 이 LineLeftFragment, LineRightFragment 안에 이전 결과 내역을 알 수 있는 measure 값을 그래프로 나타내는 Fragment와 List 영역의 Fragment로 나뉜다. LineBothFragment에서는 그래프 결과 값없이 그림 리스트로만 확인이 가능하다.

LineRight_Fragment, LineLeft_Fragment : 이전 검사 결과를 알 수 있는 measure 값을 그래프로 나타내는 Fragment이다. 각 6개의 탭마다 Fragment가 존재하여, 그래프 값을 확인할 수 있다.

TotalFragment, HzFragment, MagnitudeFragment, DistanceFragment, SpeedFragment, TimeFragment : 각 measure 별로 결과 값을 그래프로 알려준다. 파이어베이스에서 값을 실시간으로 받아와서 그래프를 생성한다.

Line_List_Fragment, Line_Rectangle_Fragment, Line_Both_Rectangle_Fragment : 일반 List와 그림 List는 각 다른 RecyclerView Adapter를 사용한다. Line_Rectangle_Fragment 안에는 왼손, 양손 RecyclerView를 따로 사용하여 그림이 나타나도록 한다.



Task_Select : SpiralTask_Fragment와 LineTask_Fragment에서 '새로운 검사 시작하기' 버튼을 눌렀을 시, 나선 검사와 선 검사인지 구분하여 왼손 검사를 실행할건지 오른손 검사를 실행할건지 선택하는 Activity이다. 선택을 하고 난 후, 나선 검사일 때, 'Spiral'로 선 검사일 때, 'Line' Activity로 이동한다.

SpiralResultActivity : 검사를 완료하고 난 후, 해당 날짜로 리스를 클릭할시 나타나는 결과 Activity. 그 린 나선 이미지와 각 measure의 그래프 결과와 이전 결과 추이를 확인하는 Fragment가 있다.

LineResultActivity : 검사를 완료하고 난 후, 해당 날짜로 리스를 클릭할시 나타나는 결과 Activity. 그 린 선 이미지와 각 measure의 그래프 결과와 이전 결과 추이를 확인하는 Fragment가 있다.

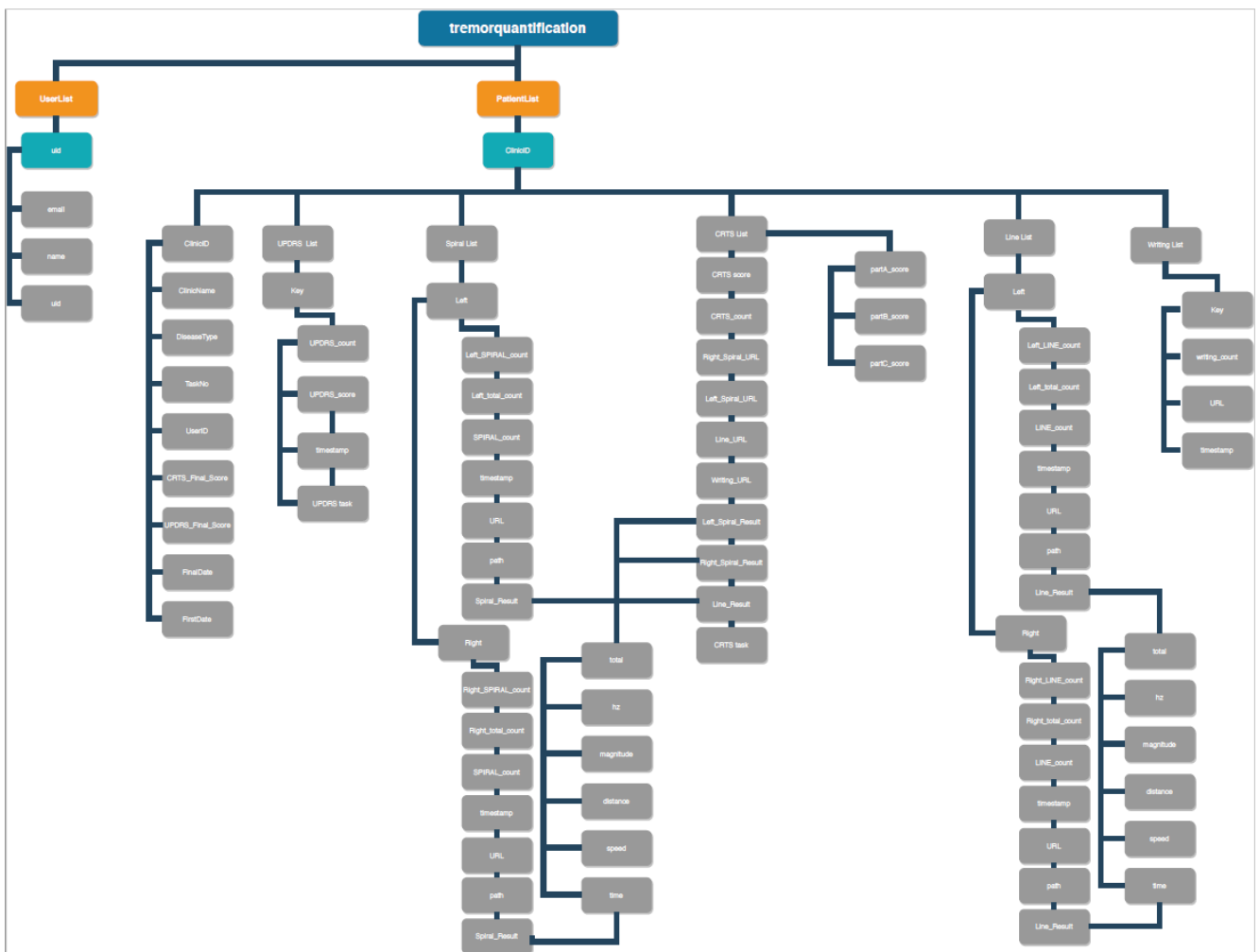
Writing_Fragment : CRTS 검사 진행 시 실시한 글씨 검사의 결과를 확인할 수 있다. 일반 List와 그림 List 둘 다 확인할 수 있도록 Fragment가 분리되어 있다.

Writing_List_Fragment : 일반 List를 나타내는 Fragment이다. 리스트 클릭시 WritingResult로 이동한다.

Writing_Rectangle_Fragment : 그림 List를 나타내는 Fragment이다. 리스트 클릭시 WritingResult로 이 동한다.

WritingResult : 파이어베이스에 저장된 그림 이미지를 불러오도록 한다.

7 데이터 베이스 구조



UserList : 구글 계정의 정보를 담는 List

Key name	Description
uid	로그인한 Google 계정의 uid를 key로 가짐.

Data name	Data Type	Description
uid	String	Google uid
email	String	Google email
name	String	Google name

PatientList : 등록된 환자의 정보를 담는 List

Key name	Description
ClinicID	등록한 환자 ID를 key로 가짐.

Data name	Data Type	Description
ClinicID	String	등록한 환자 ID
ClinicName	String	등록한 환자 이름
CRTS_Final_Score	int	CRTS 검사 마지막 점수 기록
DiseaseType	String	CRTS 검사를 먼저 한 경우 - P(Parkinson). UPDRS 검사를 먼저 한 경우 - ET(Essential tremor).
FinalDate	Date	마지막 검사 날짜
FirstDate	Date	처음 검사 날짜
TaskNo	int	검사 실시 횟수
UPDRS_Final_Score	int	UPDRS 검사 마지막 점수 기록
UserID	String	UserList의 uid 구글 계정 마다 환자 관리 가능
UPDRS List		UPDRS 검사 결과 저장 List
CRTS List		CRTS 검사 결과 저장 List
Spiral List		Spiral 검사 결과 저장 List
Line List		Line 검사 결과 저장 List
Writing List		Writing 검사 결과 저장 List

UPDRS List

Key name	Description
Key value	중복 없이 firebase 자체적으로 생성하는 key

Data name	Data Type	Description
UPDRS_count	int	UPDRS 검사 횟수 기록.
UPDRS_score	int	UPDRS 검사 결과 기록.

timestamp	Date	UPDRS 검사 날짜와 시간 기록.
UPDRS task	int	UPDRS 각 검사 항목의 값

CRTS List

Key name	Description
Key value	중복 없이 firebase 자체적으로 생성하는 key

Data name	Data Type	Description
CRTS_count	int	CRTS 검사 횟수 기록.
CRTS score	int	Part A, B, C 검사 결과 기록
timestamp	Date	CRTS 검사 날짜와 시간 기록.
Right_Spiral_URL	String	오른손 나선 검사 이미지 URL
Left_Spiral_URL	String	왼손 나선 검사 이미지 URL
Line URL	String	선 검사 이미지 URL
Writing_URL	String	쓰기 검사 이미지 URL
CRTS task	int	CRTS 각 검사 항목의 값
Right_Spiral_Result	double	오른손 나선 검사 결과 (5가지)
Left_Spiral_Result	double	왼손 나선 검사 결과 (5가지)
Line_Result	double	선 검사 결과 (5가지)

Right_Spiral_Result, Left_Spiral_Result, Line_Result(CRTS List), Spiral_Result(Spiral List), Line_Result(Line List)

Key name	Description
Result	각 Result 이름을 key로 가짐.

Data name	Data Type	Description
distance	double	기준선에서 오차 거리
hz	double	떨림의 세기
magnitude	double	떨림의 빈도
speed	double	속도
time	double	시간
total	double	

Spiral List – Left

Key name	Description
Key value	중복 없이 firebase 자체적으로 생성하는 key

Data name	Data Type	Description
Left_SPIRAL_count	int	왼손 나선 검사 횟수

Left_total_count	int	전체(crts, spiral) 왼손 나선 검사 횟수
path	String	Spiral 검사, CRTS 검사 구분
SPIRAL_count	int	전체 나선 검사 횟수
Spiral_Result	double	왼손 나선 검사 결과 (5가지)
timestamp	Date	검사 날짜와 시간 기록.
URL	String	왼손 나선 검사 이미지 URL

Spiral List – Right

Key name	Description
Key value	중복 없이 firebase 자체적으로 생성하는 key

Data name	Data Type	Description
Right_SPIRAL_count	int	오른손 나선 검사 횟수
Right_total_count	int	전체(crts, spiral) 오른손 나선 검사 횟수
path	String	Spiral 검사, CRTS 검사 구분
SPIRAL_count	int	전체 나선 검사 횟수
Spiral_Result	double	오른손 나선 검사 결과 (5가지)
timestamp	Date	검사 날짜와 시간 기록.
URL	String	오른손 나선 검사 이미지 URL

Line List – Left

Key name	Description
Key value	중복 없이 firebase 자체적으로 생성하는 key

Data name	Data Type	Description
Left_LINE_count	int	왼손 선 검사 횟수
Left_total_count	int	전체(crts, line) 왼손 선 검사 횟수
path	String	Line 검사, CRTS 검사 구분
LINE_count	int	전체 선 검사 횟수
Line_Result	double	왼손 선 검사 결과 (5가지)
timestamp	Date	검사 날짜와 시간 기록.
URL	String	왼손 선 검사 이미지 URL

Line List – Right

Key name	Description
Key value	중복 없이 firebase 자체적으로 생성하는 key

Data name	Data Type	Description
Right_LINE_count	int	오른손 선 검사 횟수
Left_total_count	int	전체(crts, line) 오른손 선 검사 횟수
path	String	Line 검사, CRTS 검사 구분
LINE_count	int	전체 선 검사 횟수
Line_Result	double	오른손 선 검사 결과 (5가지)

timestamp	Date	검사 날짜와 시간 기록.
URL	String	오른손 선 검사 이미지 URL

Writing List

Key name	Description
Key value	중복 없이 firebase 자체적으로 생성하는 key

Data name	Data Type	Description
writing_count	int	쓰기 검사 횟수
URL	String	쓰기 검사 횟수.
timestamp	String	검사 날짜와 시간 기록