AutoDistanceDriven 개요

# 기능 정리(빨간색 필수 구현)

1. 앱 실행 시 권한 요청
   1. 권한 요청 수락 시 로그인 화면으로
   2. 권한 요청 거부 시 alert 띄우고 앱 종료
2. 로그인 / 로그아웃
   1. 로그인
      1. 사용자에게 id/pw 받아 로그인 요청
      2. 서버의 응답
         1. 성공
            1. 로그인 후 메인 화면으로 이동
         2. 실패
            1. Alert 후 이동 없음
   2. 로그아웃
      1. 후에 알려주심
3. 주행 측정
   1. 메인 화면에 제작
   2. Start 버튼을 누를 시 주행측정시작
      1. 3초단위로 gps 기록
      2. 주행 중 속도, 이동시간, 총주행 거리, 종료버튼 보여주기

//주행이 종료되지 않은 상태에서(종료버튼 미선택 시) 앱을 종료해도 계속 주행 기록이 되어야한다. -> 어떻게 처리할 것인가

//주행이 종료되지 않은 상태에서 앱을 재실행하면 주행 중 상태가 유지

* 1. Finish 버튼을 누르면 주행 측정 종료 및 저장
     1. 저장 되는 것
        1. gps 주행 기록 저장
        2. 일, 요일
        3. 시작시간, 종료시간, 총 이동시간, 총 이동거리

1. 주행거리 저장
   1. 메인화면에서 이동 가능(알아서)
   2. 시작시간, 종료시간, 총이동시간, 총이동거리 항목 (주행 기록 목록) 확인 가능
   3. 주행 기록 목록을 선택하면 상세로 이동하고 구글 맵에 이동경로를 마킹

# 조건

* kotlin(언어)
* mvvm(아키텍쳐)
* coroutine(백그라운드처리 (네트워크 리퀘스트 (Retrofit, Volley 등) 내부 저장소 접근 (Room, SQLite))
* koin(의존성 주입)
* okhttp, retrofit(서버 통신)
* room(DB)
* 주행 기록 데이터는 local db에 저장
* minSdkVersion=21, targetSdkVersion=30

# 개발 기간별 기획

* 프로젝트 설계
  + DB 설계(주행 기록 데이터) => 레포지토리
  + 화면 설계
  + MVVM 설계
    - 모델 설계
    - 뷰 설계 -> navigation 사용?
    - 뷰 모델 설계
  + 기능 개발 설계
* UI 개발 -> 디자인은 뒤로 넣기
  + 주행 기록 보는 화면 설계 시 기간
* 기능 개발
  + 백그라운드 처리 주의 => 데이터 보존 필요

# DB 설계

* 필요한 디비
* 필요한 쿼리

# MVVM 설계

* 필요한 모델(DB), 뷰모델, 뷰, 레포지토리 생각하기
* 어떤 라이브러리를 사용할 것 인가?

# 화면 설계

* Navigation 사용?
* 액티비티 2개(로그인, 메인)
* 로그인 액티비티
* 메인 액티비티
* 필요한 뷰모델 생각 잘하기

# 평가 발표 내용

* 사용 기술 스택
  + 기술 스택 정리하기
* 설계 리뷰
  + DB
  + MVVM에서 구조 잡은 이유
  + 설계 단계
* 개발 중 주요 이슈
  + 개발 막힐 때 마다 메모
  + 이슈 관리 잘하기
* 시연

# 평가 기준

* 요구사항의 기능들이 모두 구현되었는가? (50%)
* 팀원들이 읽기 편한 코드로 작성되었는가? PR 리뷰가 반영되었는가?(30%)
* 가이드 된 기술 스택이 사용 되었는가?(20%)

# 고민되는 문제점

* 주행거리를 어떻게 계산 할 것인가
  + 시작과 끝은 안됨
  + 3초마다 거리 계산 후 더한다 => 주행거리 실시간으로 변하니까 이게 맞을 듯
  + 실시간으로 업데이트가 필요하다. (백그라운드 처리 고민)
  + 디비에 처음 위치, 마지막 위치 위도 경도가 필요할까? => 위치(동까지)로 저장
* 어떻게 사용자의 위치를 추적 할 것인가?
* 어떻게 이동경로를 저장할 것인가?
  + 사용자가 누구인지 저장 필요 User DB 필요 -> userID와 id만 저장
  + 실시간 이동경로를 위한 DB 하나
  + 주행거리, 주행시간, 시작 시간, 종료시간 DB 하나
  + 달로 나눠줄 수 있는 쿼리 필요?
  + 3초마다의 기록 저장 기준은? => 무조건
    - <https://developers.google.com/maps/documentation/android-sdk/polygon-tutorial?hl=ko#kotlin_2> -> 맵에 선 그리기
    - 이전의 기록에서 ?M 이상 움직였을 때 유의미 할 듯 -> 무슨 기준으로 판단?-> 속도?
  + 3초마다 어떻게 기록을 저장 할 것인가….
* 구글 키 보안
* Chrometer를 사용할 것인가 timer를 사용한 textview를 사용할 것인가
  + 데이터의 관리를 위해 textview사용(viewModel이 독립적이기 위해서)
* 화면 설계에서 리사이클러뷰 안의 리사이클러뷰를 할 것인가
  + DB에서 날짜만 뽑아오는 것 성공하면 시도 아니면 plan1로
* 이 프로젝트에서 Koin을 어디에 적용시킬 수 있는가?
  + 코인이란..? 왜 사용하지?
* 마지막 행의 id 가져오는 법 DB Select 작동원리

# 참조

* <https://vagabond95.me/posts/android-pakage-structure/> - 패키지구조
* <https://nhj12311.tistory.com/393> - 포스트맨 사용법
* <https://araikuma.tistory.com/761> - sqlite 마지막행
* <https://developer.android.com/training/location/request-updates?hl=ko> –위치 정보 부르기
* <https://developer.android.com/training/location/background?hl=ko> – 백그라운드 위치처리
* <https://bitsoul.tistory.com/131> - 위치정보 가져오기
* <https://jhappy.tistory.com/entry/Scheduling-Repeating-Alarms> - 알람메니저 설정방법
* <https://ddangeun.tistory.com/124> - 블루투스 Service 사용법
* <http://koreascience.or.kr/article/CFKO201835146901019.pdf> - SQLite 3 insert 비교 논문