

Walking Mate

project.D

2016314670 문정연

2014314143 김규용

2018311304 성보경

2016311561 송유호

Contents

1 Front-End

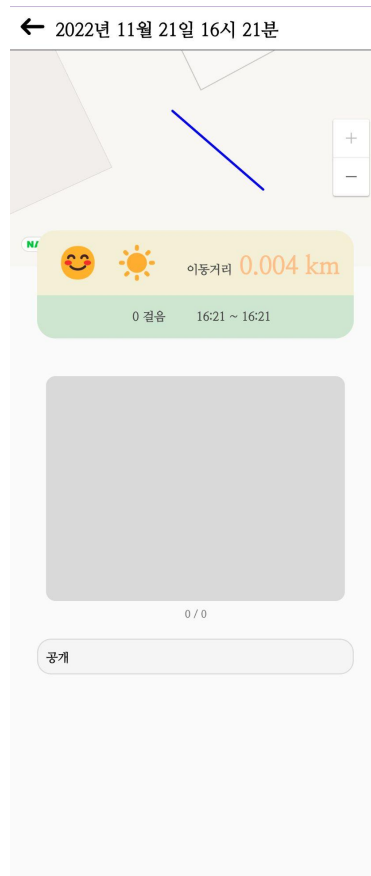
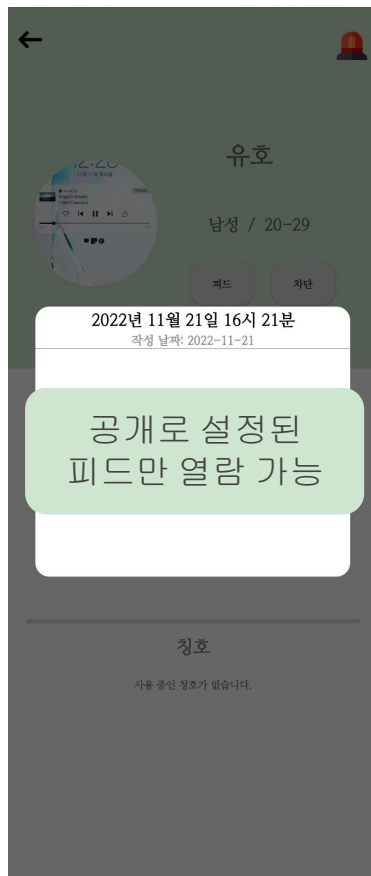
2 Back-End

3 Milestone

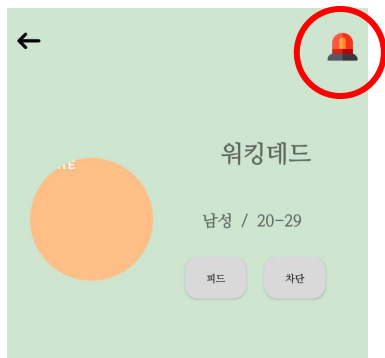


Front-End

Implementation - 유저 프로필



Implementation - 유저 프로필

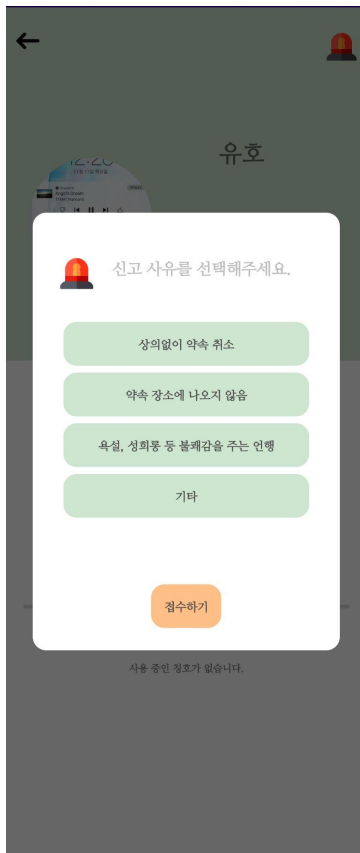


신뢰도

50

칭호

사용 중인 정보가 없습니다.



신고 사유를 선택해주세요.

상의없이 약속 취소

약속 장소에 나오지 않음

욕설, 성희롱 등 불쾌감을 주는 언행

기타

접수하기

사용 중인 정보가 없습니다.



신고 사유를 선택해주세요.

상의없이 약속 취소

약속 장소에 나오지 않음

욕설, 성희롱 등 불쾌감을 주는 언행

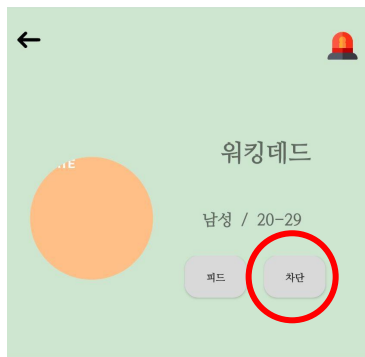
기타

신고사유를 적어주세요.

접수하기

사용 중인 정보가 없습니다.

Implementation - 유저 프로필

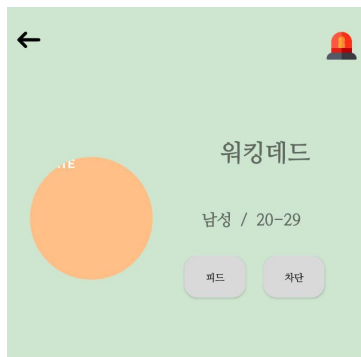


신뢰도



칭호

사용 중인 칭호가 없습니다.

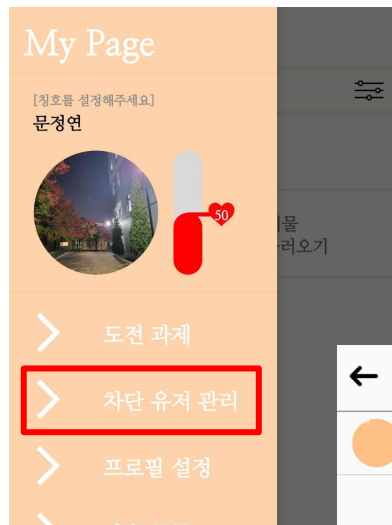
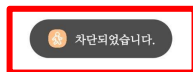


신뢰도



칭호

사용 중인 칭호가 없습니다.



← 차단 유저 관리



워킹데드 (남성/20-29)

차단 해제

이미 차단한 유저일 경우
=> “이미 차단한 유저입니다.”

중복 신고 방지 방안

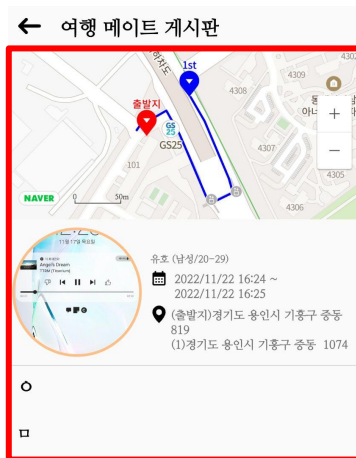
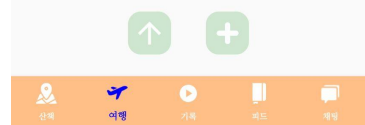
특정 유저의 프로필을 불러올 때 해당 유저와 사용자의 **id**를 모두 활용한다. 신고할 때 이 두 **id**의 데이터를 체크하여 신고 전적을 확인함으로써 중복 신고를 방지할 수 있다.

Implementation - 여행 게시판



writetime 기준
내림차순으로 **50개씩**
게시물을 받아오고, 해당
칸을 클릭시 추가로
50개씩 더 불러오는 방식

단, 내 게시물 보기
모드에서는 게시물
제한이 없음



내 게시물일 경우
=> “신청자 확인하기”로 변경

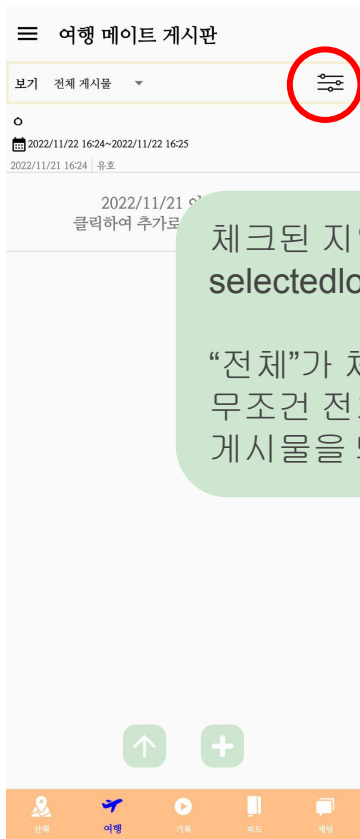
전체 경로를 띄우기 위해 루트 좌표set을
기반으로 지도가 비추는 범위를 설정

프로필 이미지(big), 이름, 성별, 나이대

이미 신청한 게시물일 경우
=> “요청한 게시물입니다.”

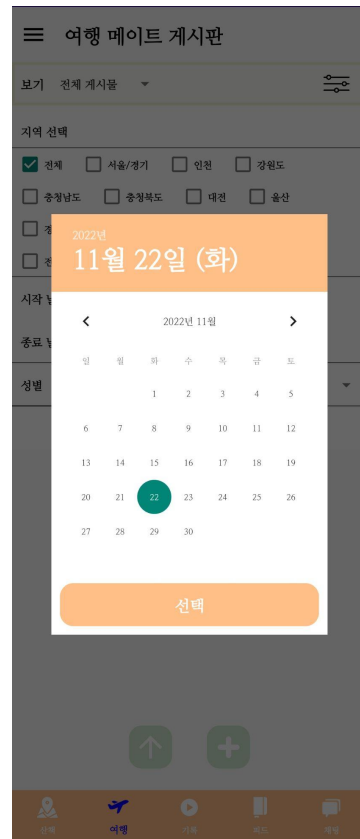
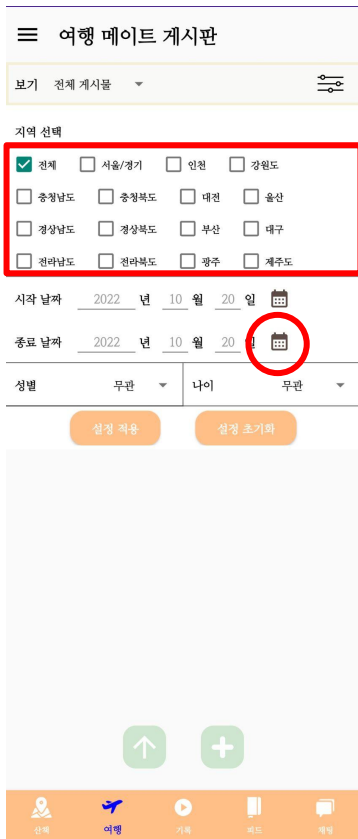
삭제된 게시물일 경우
=> “삭제된 게시물입니다.”

Implementation - 여행 게시판



체크된 지역은
selectedlocation에 추가

“전체”가 체크되어 있으면
무조건 전체 지역
게시물을 띄움



Implementation - 여행 게시판

원소 “last”

전체 게시물 보기에서 last: “yyyy/mm/dd 이전 게시물 \n 클릭하여 추가로 게시물 불러오기”

내 게시물 보기에서 last: “게시물이 존재하지 않습니다.”

신청 게시물 보기에서 last: “신청한 게시물이 존재하지 않습니다.”

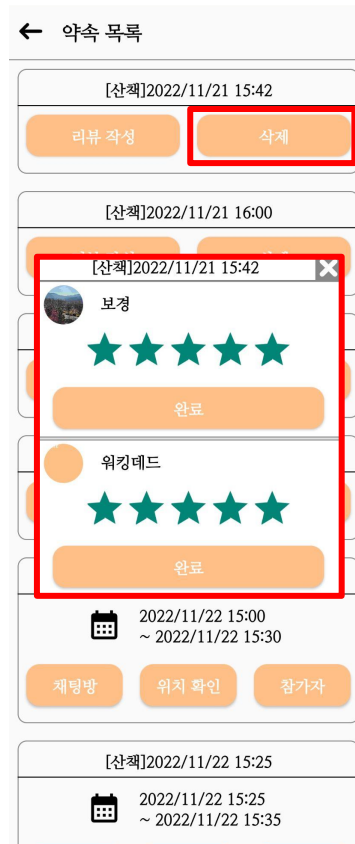
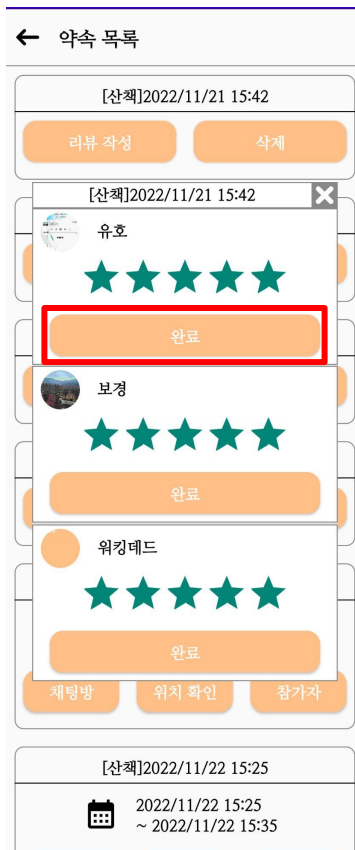
swiperefreshlayout

리스트 부분을 잡아 당겨 새로고침

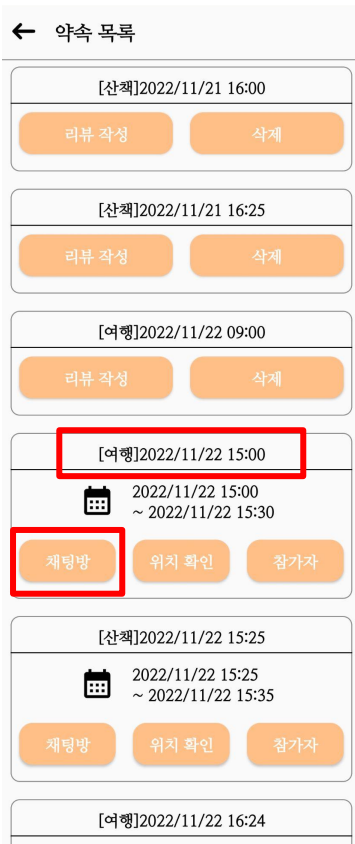
파이어베이스 쿼리 지원

파이어베이스가 복잡한 쿼리를 지원하지 않기 때문에 지역 선택과 성별, 연령대, 작성 시간만 쿼리를 적용하고 나머지는 어플 자체에서 처리한다.

Implementation - 약속 리스트



Implementation - 약속 리스트



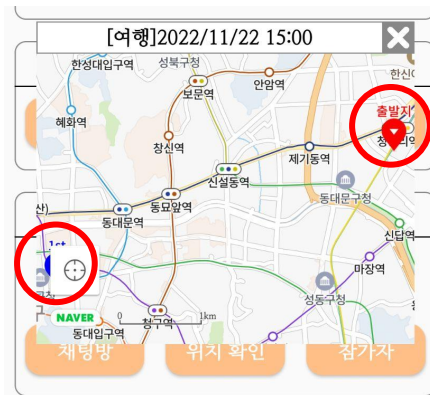
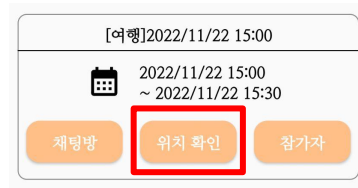
← [여행][수락]2022/11/22 15:00

내가 들어있는 채팅방
=> 해당 채팅방을 열어줌

채팅방은 존재하지만
내가 나온 경우
=> 다시 들어가게 해줌

채팅방 자체가 존재하지
않는 경우
=> “더이상 존재하지
않는 채팅방입니다.”

채팅 입력



루트 없이 마커만 표시

Implementation - 약속 리스트

← 약속 목록

[산책]2022/11/21 16:00

리뷰 작성 삭제

[산책]2022/11/21 16:25

리뷰 작성 삭제

[여행]2022/11/22 09:00

리뷰 작성 삭제

[여행]2022/11/22 15:00

2022/11/22 15:00
~ 2022/11/22 15:30

채팅방 위치 확인 참가자

[산책]2022/11/22 15:25

2022/11/22 15:25
~ 2022/11/22 15:35

채팅방 위치 확인 참가자

[여행]2022/11/22 16:24

← 약속 목록

[산책]2022/11/21 16:00

리뷰 작성 삭제

[산책]2022/11/21 16:25

리뷰 작성 삭제

[여행]2022/11/22 15:00

보경
문정연
위경데드
유호

채팅방 위치 확인 참가자

[산책]2022/11/22 15:25

2022/11/22 15:25
~ 2022/11/22 15:35

채팅방 위치 확인 참가자

[여행]2022/11/22 16:24

약속 리스트 정렬 방법

약속의 시작 시간을 기준으로 비교해서 정렬한다. 단, 종료된 약속들을 최상단으로 배치한다.

별점 => 신뢰도 변환 방식

별점에 따른 신뢰도 점수를 약속 참여 유저수로 나누어 계산한다. 이는 많은 유저가 참여한 약속에서 한꺼번에 많은 신뢰도를 얻는 것을 방지하기 위함이다. 100점을 초과하면 100점으로, 0점 미만이 되면 0점으로 표시한다.

아래는 신뢰도 점수 환산 기준이다.

0~1점 -3점

1점 초과 2점 이하 -2점

2점 초과 3점 이하 -1점

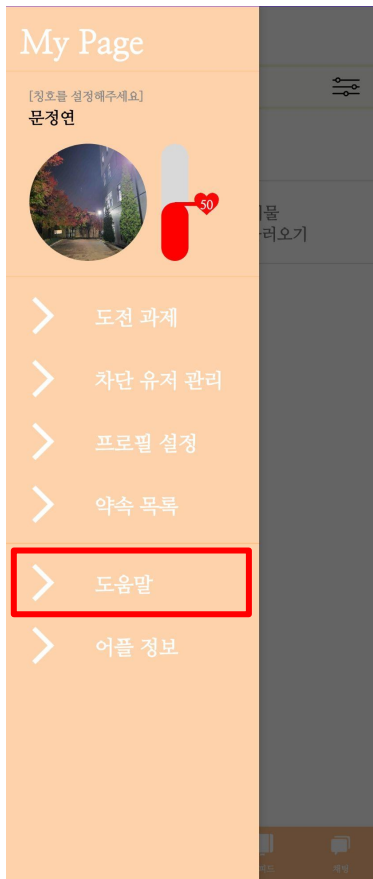
3점 초과 4점 이하 유지

4.5점 +1

5점 +2

큰 불만이 없으면 사전 설정된 점수에서
깎지 않으려는 사람들의 경향성 고려

Implementation - 도움말



클릭 또는 스크롤

Back-End

Implementation - 파이어베이스 보안규칙

```
{
  "rules": {
    "roomId": {
      "uid": {
        ".read": "uid === auth.uid",
        ".write": "uid === auth.uid"
      }
    },
    "Users": {
      ".read": "auth != null",
      "uid": {
        ".read": "auth != null",
        ".write": "uid === auth.uid"
      }
    },
    "UsersConnection": {
      ".read": "auth != null",
      ".write": "auth != null"
    },
    "Messages": {
      "$roomId": {
        ".read": "root.child('RoomUsers').child($roomId).child(auth.uid).exists()",
        ".write": "root.child('RoomUsers').child($roomId).child(auth.uid).exists()",
        "$msgId": {
          ".read": "root.child('RoomUsers').child($roomId).child(auth.uid).exists()",
          ".write": "root.child('RoomUsers').child($roomId).child(auth.uid).exists()"
        }
      }
    },
    "UserRooms": {
      "uid": {
        ".read": "uid === auth.uid",
        "$roomId": {
          ".read": "root.child('RoomUsers').child($roomId).child(auth.uid).exists()",
          ".write": "root.child('RoomUsers').child($roomId).child(auth.uid).exists()"
        }
      }
    },
    "RoomUsers": {
      "$roomId": {
        "uid": {
          ".read": "auth != null",
          ".write": "auth != null"
        }
      }
    }
  }
}
```

- 인증받은 사용자만이 채팅 시스템 이용가능
- 해당 채팅방 유저만 데이터 접근 가능

참 조 : <https://firebase.google.com/docs/rules>
<https://wikidocs.net/21608>

Implementation - 파이어베이스 보안규칙

```
//users collection내에 특정 uid의 문서가 존재하는가
```

```
if exists(/databases/{database}/documents/users/{request.auth.uid})
if exists(/databases/{database}/documents/Users/{( '2s5xd0uygpMv0raIWfTxBRZcYo73' )});
```

```
//users subcollection내에 request.auth.uid인 문서가 존재하는가
```

```
if exists(/databases/{database}/documents/leagues/{(league)}/users/{request.auth.uid})
```

```
//Users 컬렉션 내 특정 문서의 'name' 필드를 write하려고 한다.
//새로운 name의 필드값이 Kits의 document id에 존재할 때만 허용한다.
```

```
match /Users/{user} {
    allow write :
        if exists(/databases/{database}/documents/Kits/{(request.resource.data.name)});
}
```

```
//Users 컬렉션 내 특정 문서의 name 필드를 read 혹은 write하려고 한다.
//write하고자 하는 문서의 birth 필드값이 Kits의 document id로 존재할 때 허용한다.
```

```
match /Users/{user} {
    allow read, write :
        if exists(/databases/{database}/documents/Kits/{(resource.data.birth)});
}
```

```
//지정된 문서의 reviewerID 필드와 request.auth.uid에 같을 때만 문서를 수정할 수 있도록 한다.
allow update: if resource.data.reviewerID == request.auth.uid
```

```
//score의 필드값을 변화시키지 않기 위함.
allow update: resource.data.score == resource.data.score;
```

```
// 지정문서의 x 필드의 값이 5 이상인 데이터만 사용자가 읽도록 하기 위함
allow read: if resource.data.x > 5;
```

```
// 지정문서의 published 필드값이 true인 데이터만 사용자가 읽도록 하기 위함
allow read: if resource.data.published == true ;
```

```
allow read: if resource.data.name == 'John Doe'
```

```
service cloud.firestore {
```

```
    match /databases/{database}/documents {
```

```
        //지정문서범위 바깥 쪽에 있는 함수
```

```
            function outerauthorOrPublished(storyid) {
                return resource.data.published == true || request.auth.uid == resource.data.author;
            }
        match /stories/{storyid} {
```

```
            //지정문서범위 내에 있는 함수
```

```
                function innerauthorOrPublished() {
                    return resource.data.published == true || request.auth.uid == resource.data.author;
                }
            }
```

```
        allow list: if request.query.limit <= 10 &&
            innerauthorOrPublished();
```

```
        allow get: if outerauthorOrPublished(storyid);
        allow write: if request.auth.uid == resource.data.author;
    }
}
```

참조 : <https://firebase.google.com/docs/rules>
<https://seizemymoment.tistory.com/11>

Implementation - 테스트

```
2022-11-21 15:05:18.649 20711-20711 ViewRootIm...tActivity] com.example.walkingmate
2022-11-21 15:05:18.852 1506-3216 AppOps pid-1506
2022-11-21 15:05:18.854 1506-3216 AppOps pid-1506
```

```
2022-11-21 15:05:18.860 20711-20711 ViewRootIm...tActivity] com.example.walkingmate
2022-11-21 15:05:18.860 20711-20711 ViewRootIm...tActivity] com.example.walkingmate
2022-11-21 15:05:18.865 20711-20711 MSHandlerLifeCycle com.example.walkingmate
```

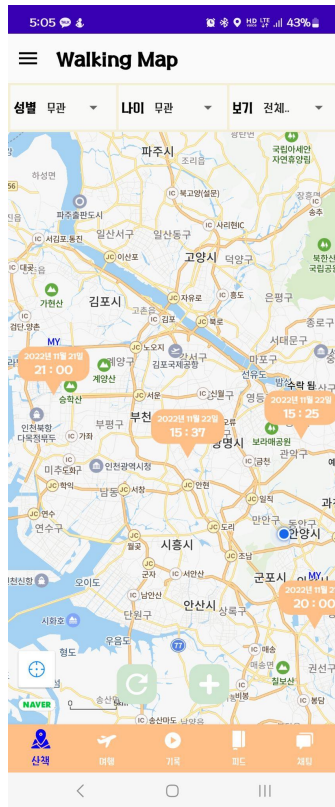
```
I Relayout returned: old=(0,0,1080,2640) new=(0,0,1080,2640) req=(1080,2640)8 dur=5 res=0x2 s={false 0x0} ch=true seqId=0
E Bad call made by uid 1000. Package "com.example.walkingmate" does not belong to uid -1.
E noteOperation
java.lang.SecurityException: Specified package "com.example.walkingmate" under uid -1 but it is not
    at com.android.server.appop.AppOpsService.verifyAndGetBypass(AppOpsService.java:4863)
    at com.android.server.appop.AppOpsService.noteOperationUnchecked(AppOpsService.java:3586)
    at com.android.server.appop.AppOpsService.noteOperationImpl(AppOpsService.java:3574)
    at com.android.server.appop.AppOpsService.-$$West$mnoteOperationImpl(Unknown Source:0)
    at com.android.server.appop.AppOpsService$CheckOpsDelegateDispatcher.lambda$noteOperation$4(AppOpsService.java:7933)
    at com.android.server.appop.AppOpsService$CheckOpsDelegateDispatcher.$r8$lambda$0wUoG558BmJ6yiwhp30Aq3tQ30(Unknown Source:0)
    at com.android.server.appop.AppOpsService$CheckOpsDelegateDispatcher$$ExternalSyntheticLambda2.apply(Unknown Source:32)
    at com.android.server.policy.AppOpsPolicy.noteOperation(AppOpsPolicy.java:227)
    at com.android.server.appop.AppOpsService$CheckOpsDelegateDispatcher.noteOperation(AppOpsService.java:7931)
    at com.android.server.appop.AppOpsService.noteOperation(AppOpsService.java:3558)
    at com.android.internal.app.IAppOpsService$Stub.onTransact(IAppOpsService.java:486)
    at android.os.Binder.execTransactInternal(Binder.java:1316)
    at android.os.Binder.execTransact(Binder.java:1280)
I stopped(true) old = false
D WindowStopped on com.example.walkingmate/com.example.walkingmate.StartActivity set to true
I removeMultiSplitHandler: no exist. decor=DecorView@33c8a28[StartActivity]
```

```
E FATAL EXCEPTION: main
```

```
Process: com.example.walkingmate, PID: 20711
```

```
java.lang.NullPointerException: Attempt to invoke interface method 'java.lang.Object java.util.Map.get(java.lang.Object)' on a null object reference
    at com.example.walkingmate.WalkFragment$10$1.onComplete(WalkFragment.java:389)
    at com.google.android.gms.tasks.zzi.run(com.google.android.gms:play-services-tasks@18.0.1:1)
    at android.os.Handler.handleCallback(Handler.java:942)
    at android.os.Handler.dispatchMessage(Handler.java:99)
    at android.os.Looper.loopOnce(Looper.java:226)
    at android.os.Looper.loop(Looper.java:313)
    at android.app.ActivityThread.main(ActivityThread.java:8772) <1 internal line>
    at com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run(RuntimeInit.java:571)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:1067)
```

Implementation - 테스트



현재 발견된 문제점

1. 장시간 이용 시 파이어베이스와 간헐적으로 연결이 끊기는 경우가 발생
- 해당 증상은 4명 중 한명만 발생한 문제점으로 원인 파악 중
2. 보안규칙과의 충돌로 인해 데이터 입출력이 안됨
- 현재는 **true**로 설정하여 입출력에 문제없음.

Milestone

구분	진행 사항	~W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15
Proposal	Proposal											
Front End	초기 디자인											
	도보 기록 및 노티 바											
	로딩 화면											
	피드 기록 및 확인(캘린더)											
	My page 사이드 바											
	여행 메이트 게시물 작성											
	산책 메이트 게시물 작성											
	피드 작성											
	하단 네비게이션 뷰 및 화면 전환											
	가입 및 유저 정보 관리(Local)											
	산책 게시판(지도)											
	프로필 페이지											
	친구 관리, 도전 과제, 도움말, 어플 정보											
	여행 게시판(리스트)											
Back End	로그인 데이터 관리											
	피드 게시물 데이터 관리											
	채팅 데이터 관리											
	산책 메이트 게시물 데이터 관리											
	여행 메이트 게시물 데이터 관리											
	여행 게시판 리스트 구현											
	유저 정보 관리											
	친구 데이터 관리											

Thank you!