2018 ITM
Mobile Programming Final

# Sports Go

14146323 Seokhyeon Chung
14146325 Hongbeom Choi

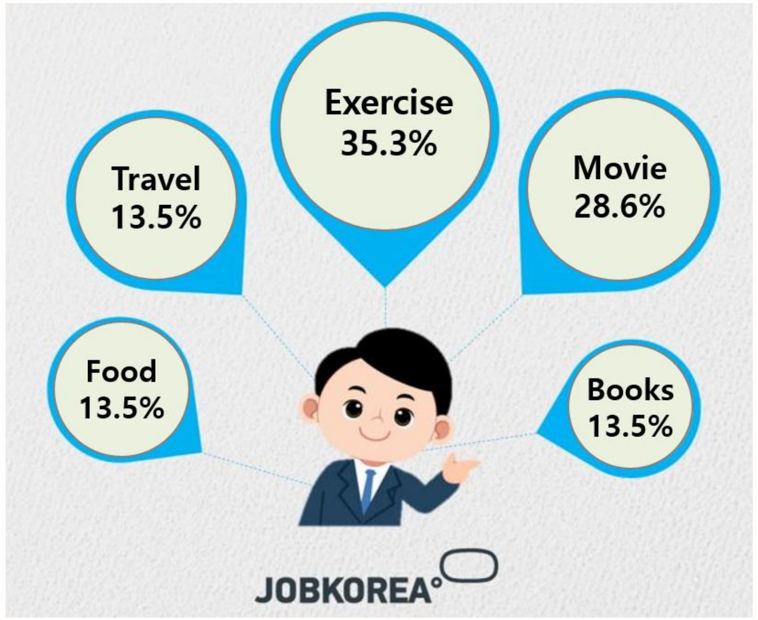
## CONTENTS

## 001 About Sports Go

- Background
- About App

## Background

### Top 5 hobbies

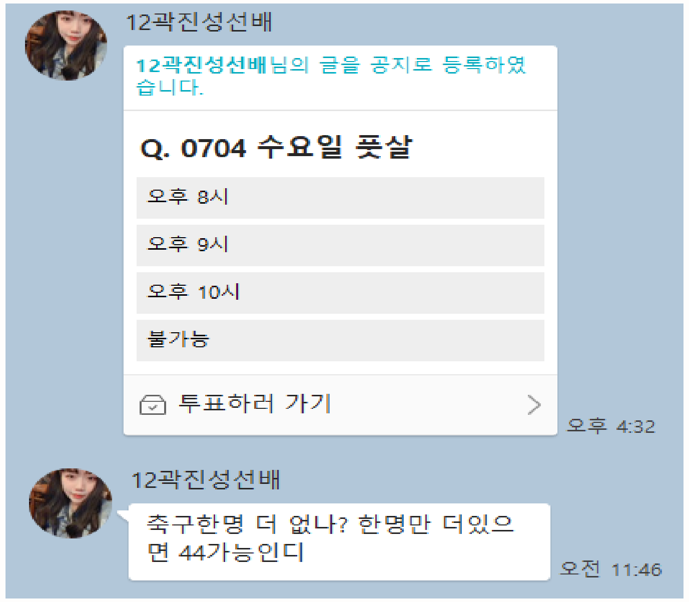


### Modern people…

- Interested in exercise
- Participate in a small exercise group with friends
- Work out at work/school or community club

## Background

## Problems



12곽진성선배

12곽진성선배님의 글을 공지로 등록하였습니다.

Q. 0704 수요일 풋살

- 오후 8시
- 오후 9시
- 오후 10시
- 불가능

투표하러 가기

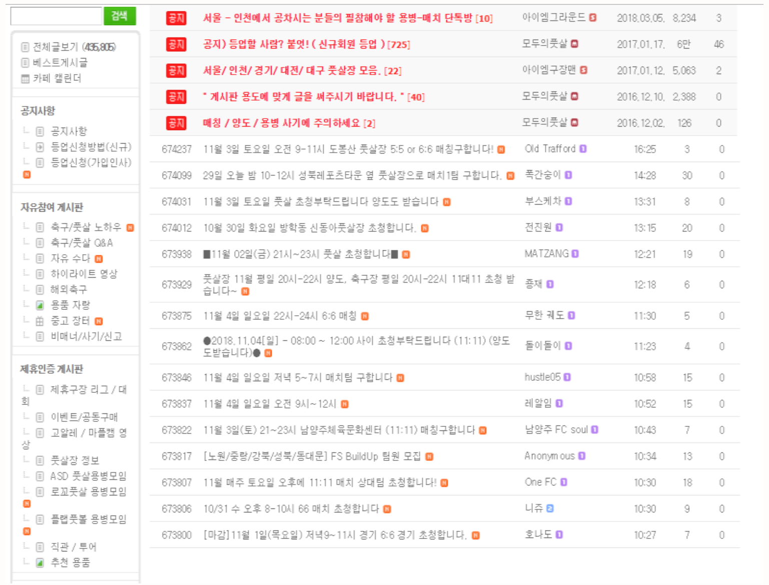
오후 4:32

12곽진성선배

축구한명 더 없나? 한명만 더있으면 44가능인디

오전 11:46

In cases where not enough people are gathered



검색

- 전체글보기 (436,806)
- 베스트게시글
- 카페 캘린더

공지사항
- 공지사항
- 등업신청방법(신규)
- 등업신청(가입인사)

자유참여 게시판
- 축구/풋살 노하우
- 축구/풋살 Q&A
- 자유 수다
- 하이라이트 영상
- 해외축구
- 용품 자랑
- 중고 장터
- 비매너/사기/신고

제휴인증 게시판
- 제휴구장 리그 / 대회
- 이벤트/공동구매
- 고알레 / 마플캠 영상
- 풋살장 정보
- ASD 풋살용병모임
- 로꼬풋살 용병모임
- 플랩풋볼 용병모임
- 직관 / 투어
- 추천 용품

| | | | | | |
|---|---|---|---|---|---|
| 공지 | 서울 - 인천에서 공차시는 분들의 필참해야 할 용병-매치 단톡방 [10] | 아이엠그라운드 | 2018.03.05. | 8,234 | 3 |
| 공지 | 공지) 등업할 사람? 붙엇! ( 신규회원 등업 ) [725] | 모두의풋살 | 2017.01.17. | 6만 | 46 |
| 공지 | 서울/ 인천/ 경기/ 대전/ 대구 풋살장 모음. [22] | 아이엠구장맨 | 2017.01.12. | 5,063 | 2 |
| 공지 | " 게시판 용도에 맞게 글을 써주시기 바랍니다. " [40] | 모두의풋살 | 2016.12.10. | 2,388 | 0 |
| 공지 | 매칭 / 양도 / 용병 사기에 주의하세요 [2] | 모두의풋살 | 2016.12.02. | 126 | 0 |
| 674237 | 11월 3일 토요일 오전 9-11시 도봉산 풋살장 5:5 or 6:6 매칭구합니다! | Old Trafford | 16:25 | 3 | 0 |
| 674099 | 29일 오늘 밤 10-12시 성북레포츠타운 옆 풋살장으로 매치1팀 구합니다. | 폭간숭이 | 14:28 | 30 | 0 |
| 674031 | 11월 3일 토요일 풋살 초청부탁드립니다 양도도 받습니다 | 부스케차 | 13:31 | 8 | 0 |
| 674012 | 10월 30일 화요일 방학동 신동아풋살장 초청합니다. | 전진원 | 13:15 | 20 | 0 |
| 673938 | ■11월 02일(금) 21시~23시 풋살 초청합니다■ | MATZANG | 12:21 | 19 | 0 |
| 673929 | 풋살장 11월 평일 20시-22시 양도, 축구장 평일 20시-22시 11대11 초청 받습니다~ | 종재 | 12:18 | 6 | 0 |
| 673875 | 11월 4일 일요일 22시-24시 6:6 매칭 | 무한 궤도 | 11:30 | 5 | 0 |
| 673862 | ●2018.11.04[일] - 08:00 ~ 12:00 사이 초청부탁드립니다 (11:11) (양도도받습니다)● | 돌이돌이 | 11:23 | 4 | 0 |
| 673846 | 11월 4일 일요일 저녁 5~7시 매치팀 구합니다 | hustle05 | 10:58 | 15 | 0 |
| 673837 | 11월 4일 일요일 오전 9시~12시 | 레알임 | 10:52 | 15 | 0 |
| 673822 | 11월 3일(토) 21~23시 남양주체육문화센터 (11:11) 매칭구합니다 | 남양주 FC soul | 10:43 | 7 | 0 |
| 673817 | [노원/중랑/강북/성북/동대문] FS BuildUp 팀원 모집 | Anonymous | 10:34 | 13 | 0 |
| 673807 | 11월 매주 토요일 오후에 11:11 매치 상대팀 초청합니다! | One FC | 10:30 | 18 | 0 |
| 673806 | 10/31 수 오후 8-10시 66 매치 초청합니다 | 니쥬 | 10:30 | 9 | 0 |
| 673800 | [마감]11월 1일(목요일) 저녁9~11시 경기 6:6 경기 초청합니다. | 호나도 | 10:27 | 7 | 0 |

Need to find the necessary number of people through Internet cafes or communities

## Background

For an app that enables everyone to enjoy sports comfortably

Realtime sports matching platform

## About…

- Sports Go is a real-time sports event matching platform
- Sports Go provides real-time information on sports events such as soccer, futsal and basketball.
- Users can participate in sports events that match their desired time/place

## 002 Market Analysis

- Target Setup
- Competitors

## Target Setup

For an app that enables everyone to enjoy sports comfortably

VS

Anyone who wants to exercise

Anyone who wants to create a team

Anyone who wants to find the opponent

## Competitor Analysis

Sports event matching platform

### N-HOUR

- N-Hour is a soccer / futsal matching platform.
- Team-only matching is possible, not individual
- User rating system and personal record do not exist.

Sports court rental platform

### I AM GROND

- I am ground is a real time sports court rental platform.
- For Soccer, futsal, basketball court information and availability.
- Matching between teams or individuals is not possible

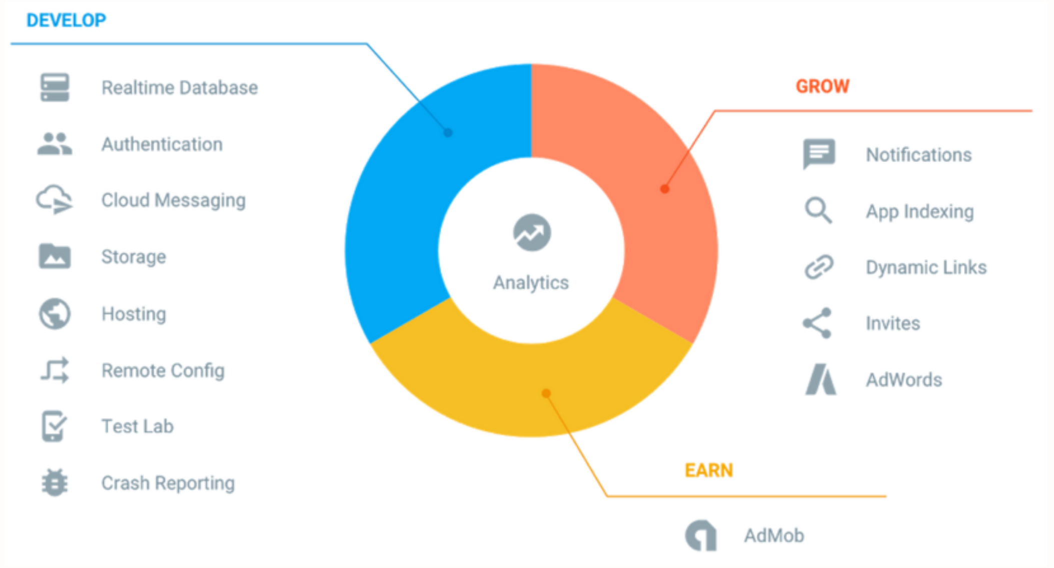
## 003 Main Functions

- Server
- UI & Scenarios

03 Main Functions

## Server

Firebase



- Baas(Backend as a Service) that provides the necessary functions for web and mobile development.
- Provides various functions like Real-time database, easy authentication, cloud storage, hosting, app testing and monetization

03 Main Functions

## UI & Scenarios

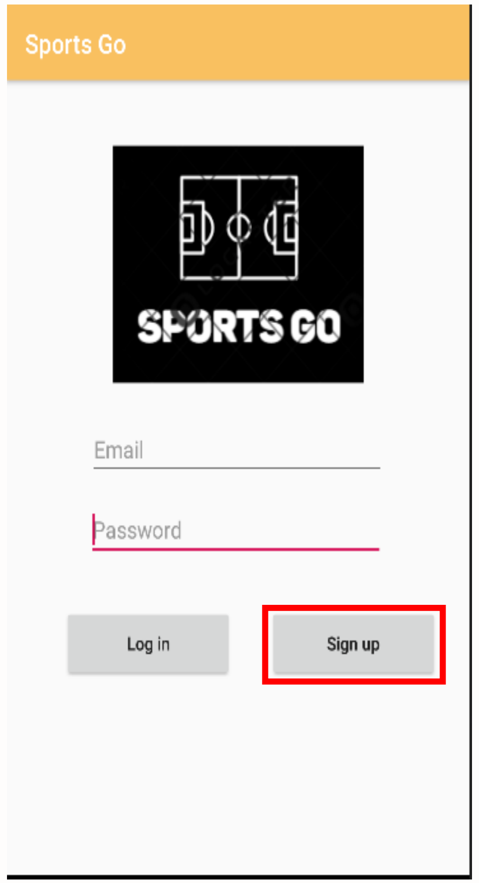
### Function List

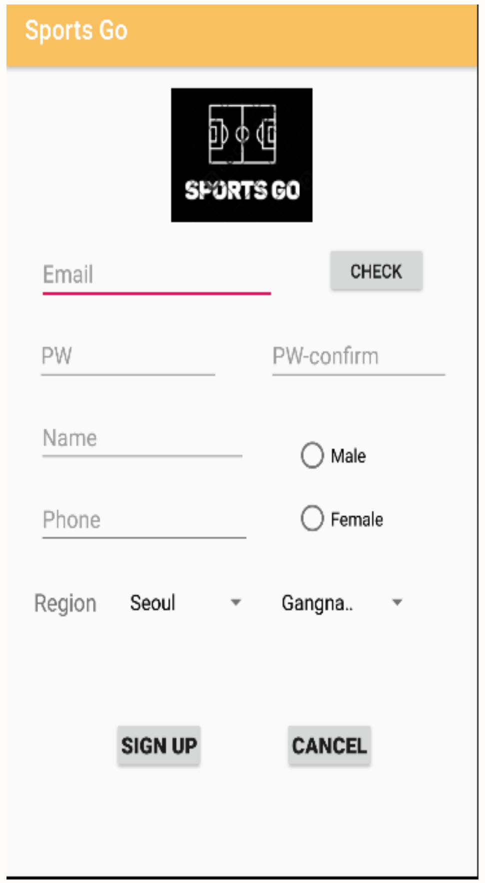
| | Function List | Planned Functions | Implemented Functions |
|---|---|---|---|
| 1 | Make an Account | O | O |
| 2 | Log-in | O | O |
| 3 | Search, Join Event | O | O |
| 4 | Create Event | O | O |
| 5 | View my event | O | O |
| 6 | Chatting | O | O |
| 7 | Nearby Facilities | O | O |
| 8 | User list and evaluate users | O | O |
| 9 | Notify match plan | O | O |
| 10 | Navigation Drawer | X | O |
| 11 | Around Event Alarm | X | O |
| 12 | Search and View location | X | O |

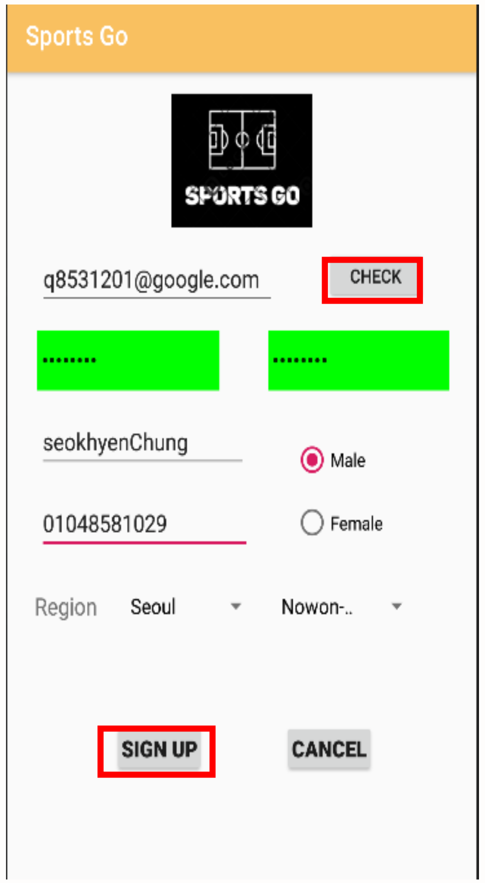
03 Main Functions

## UI & Scenarios

### 1. Sign up & Log in







q8531201@google.com

seokhyenChung

01048581029

Region Seoul Nowon-..

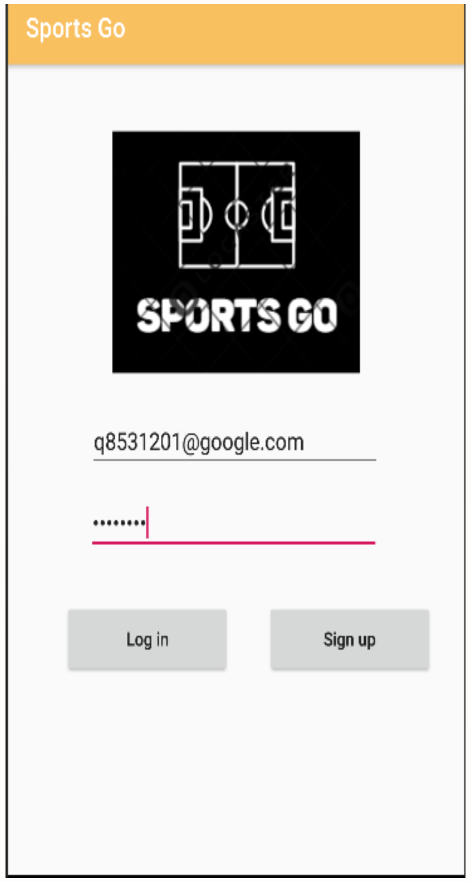
1. Click the sign up button
2. Fill out the form
3. Click the check button to verify you are a new subscriber
4. After verifying, click sign up button

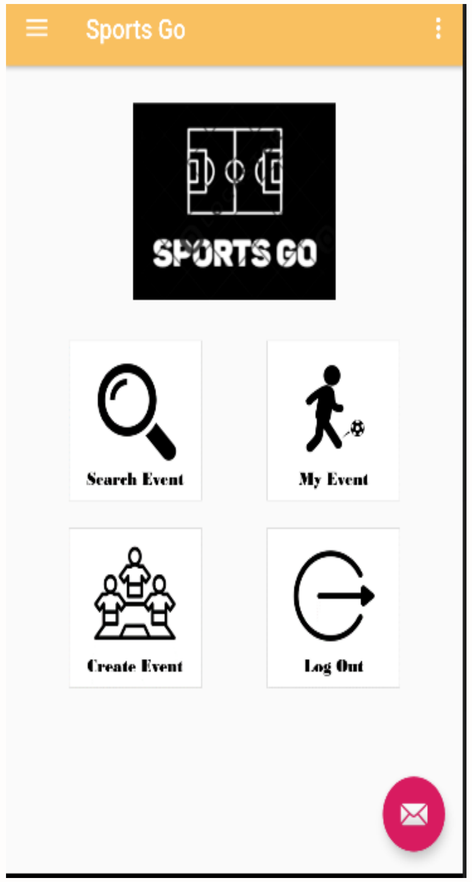
03 Main Functions

## UI & Scenarios

For an app that enables everyone to enjoy sports comfortably

### 1. Sign up & Log in

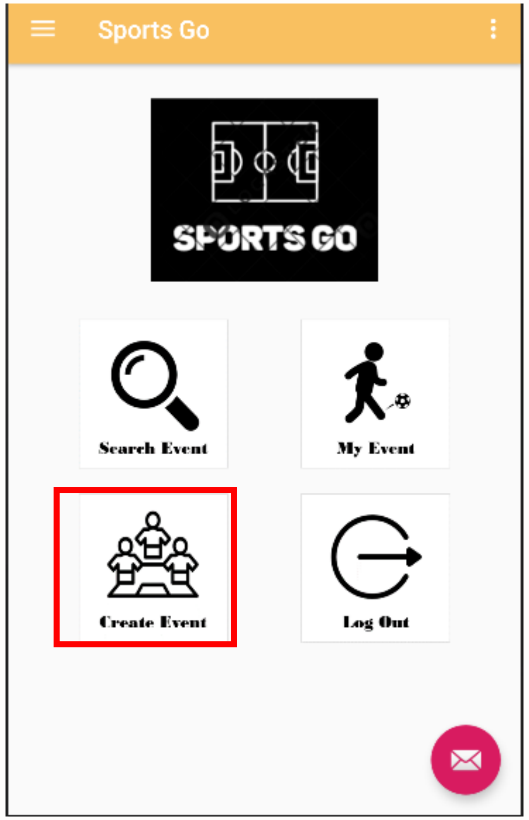


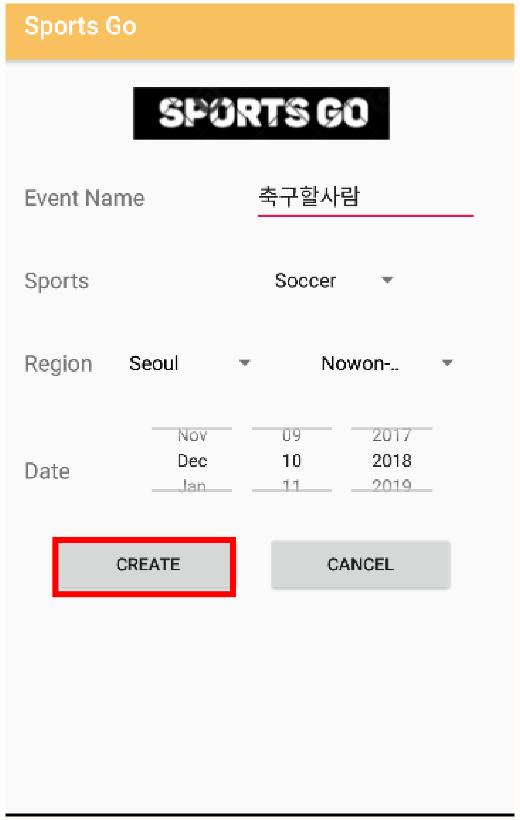


1. Enter the ID & password after signing up
2. You can log in and see the main page

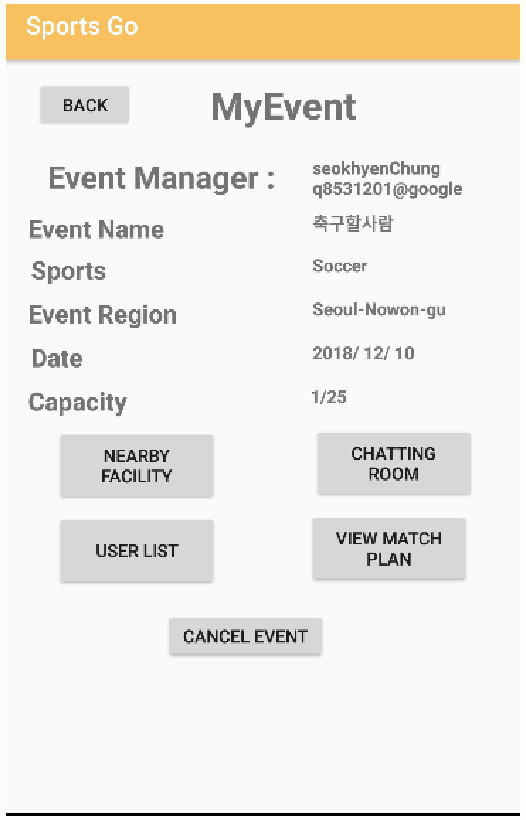
## UI & Scenarios

### 2. Create Event





| | |
|---|---|
| Event Name | 축구할사람 |
| Sports | Soccer |
| Region | Seoul Nowon-.. |
| Date | Nov Dec Jan / 09 10 11 / 2017 2018 2019 |

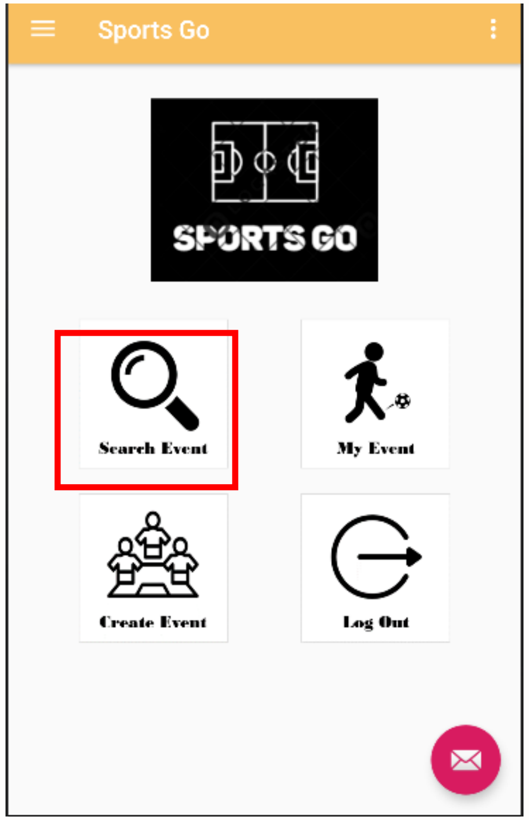


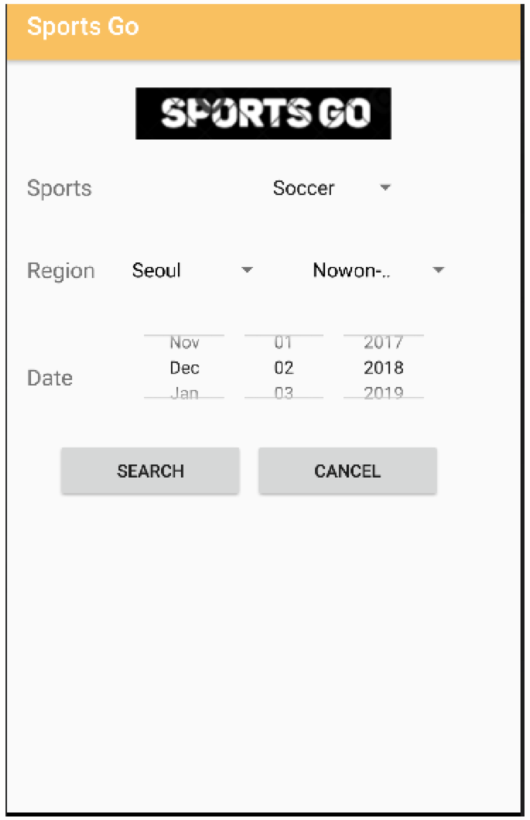
| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 1/25 |

1. Click the create event button
2. Fill out the form and click the create button
3. You can see the MyEvent page as you created and you are assigned as event manager

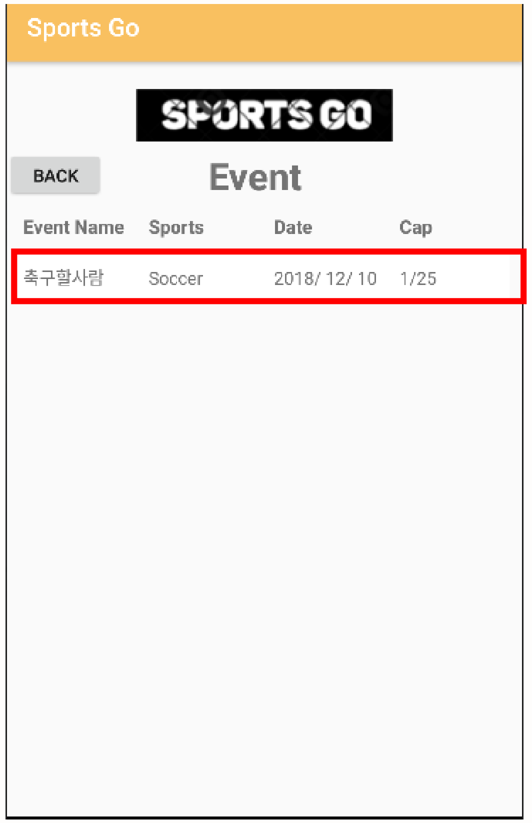
## UI & Scenarios

### 3. Search Event





Sports: Soccer
Region: Seoul, Nowon-..
Date: Dec 02 2018

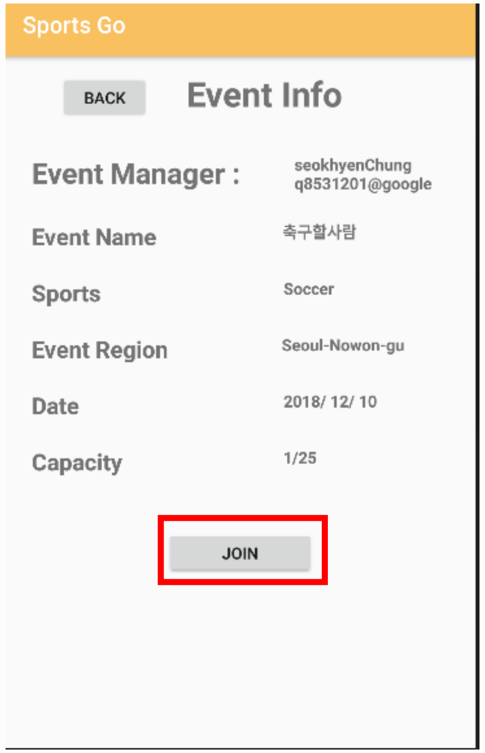


| Event Name | Sports | Date | Cap |
|---|---|---|---|
| 축구할사람 | Soccer | 2018/ 12/ 10 | 1/25 |

1. Click the search event button
2. Select the condition as you want (Sports, Region, Date)
3. You can see the searched event as your condition
4. Click the event as you want to join

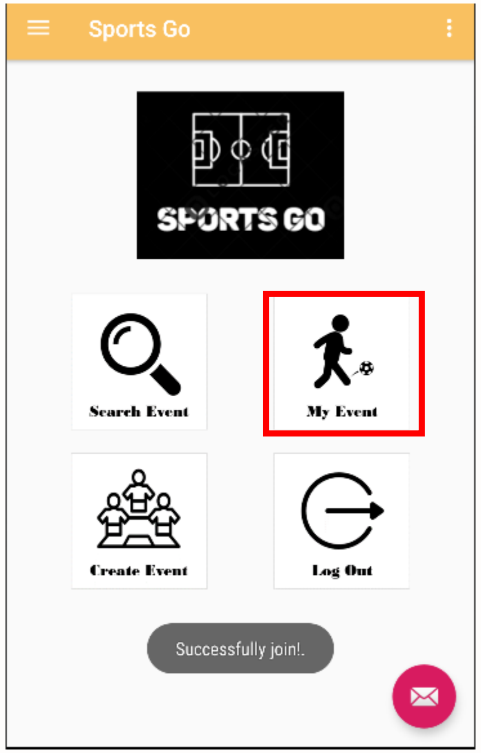
## UI & Scenarios

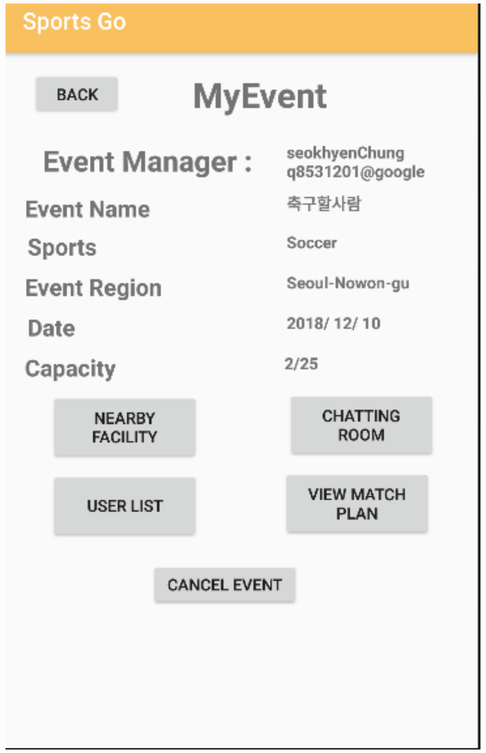
### 3. Search Event


Sports Go

BACK Event Info

| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 1/25 |




Sports Go

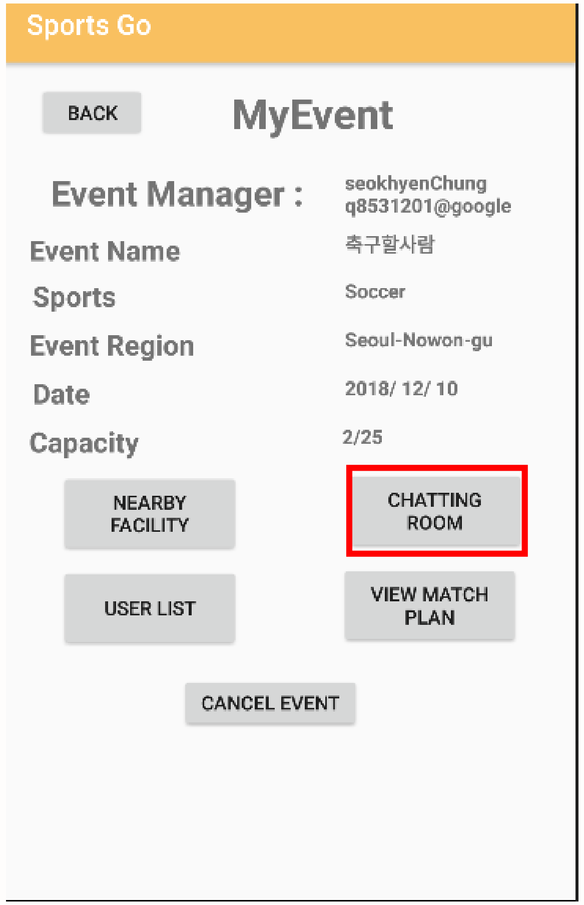
BACK MyEvent

| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 2/25 |

1. After click the event, you can see the information about the selected event
2. If you want to join, click the join button
3. You can see the joined event with clicking the My event button

## UI & Scenarios

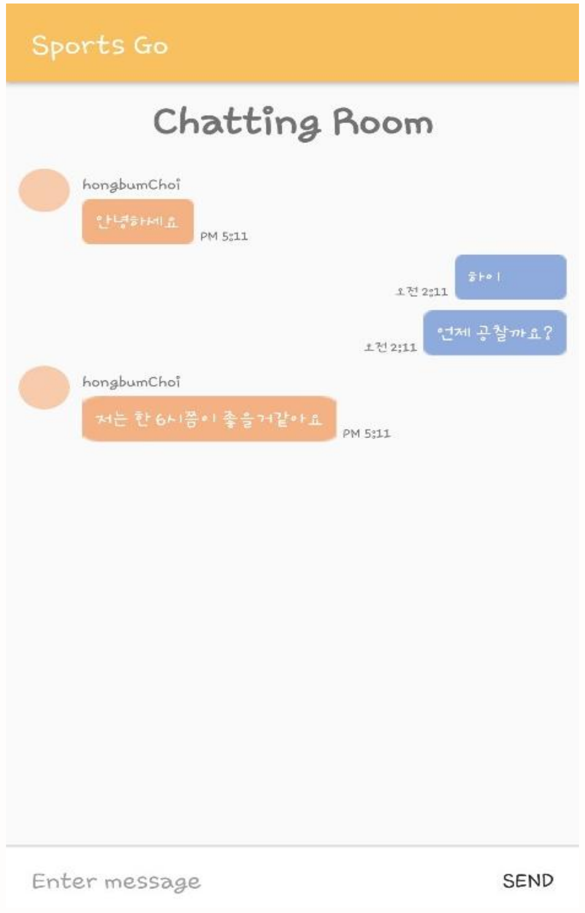
### 4. Chatting


Sports Go

BACK

MyEvent

| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 2/25 |


Sports Go

Chatting Room

hongbumChoi
안녕하세요 PM 5:11

오전 2:11 하이

오전 2:11 언제 공찰까요?

hongbumChoi
저는 한 6시쯤이 좋을거같아요 PM 5:11

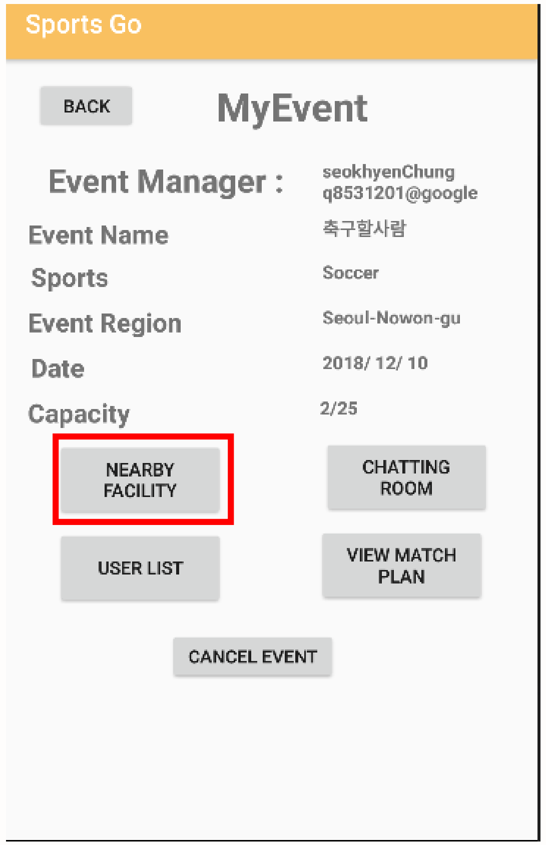
1. If you want to chat with the members, click the Chatting button

2. You can see the chatting and name of the members

3. With chatting, you can make up the plan detail

## UI & Scenarios

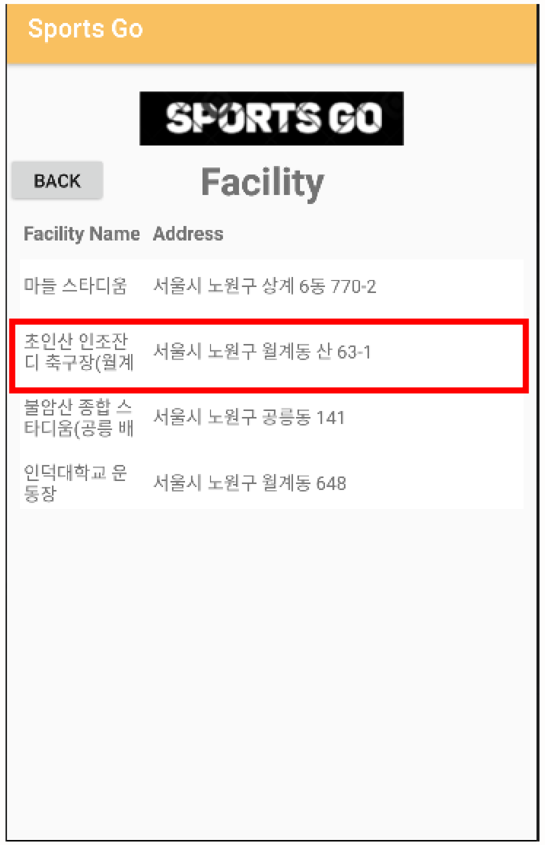
### 5. Nearby Facility



Sports Go

BACK MyEvent

| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 2/25 |



Sports Go

SPORTS GO

BACK Facility

| Facility Name | Address |
|---|---|
| 마들 스타디움 | 서울시 노원구 상계 6동 770-2 |
| 초인산 인조잔디 축구장(월계 | 서울시 노원구 월계동 산 63-1 |
| 불암산 종합 스타디움(공릉 배 | 서울시 노원구 공릉동 141 |
| 인덕대학교 운동장 | 서울시 노원구 월계동 648 |

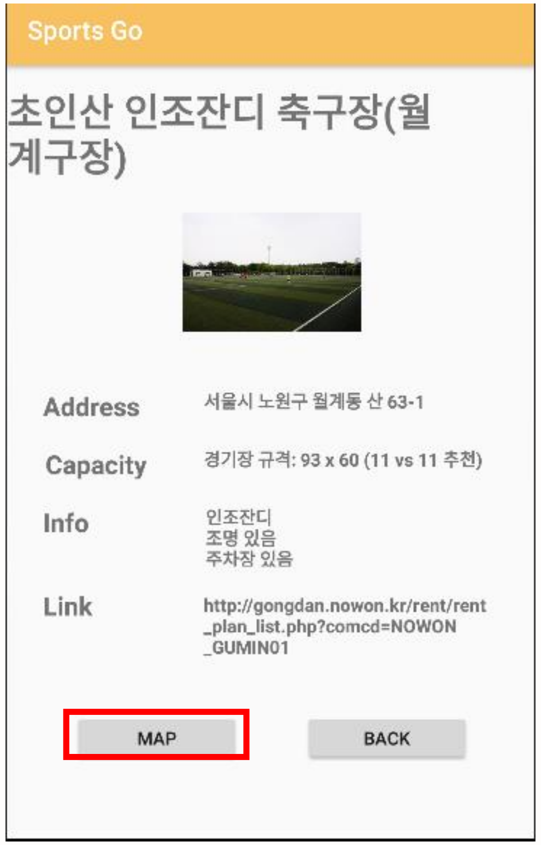
1. If you want to see the nearby facility to play sports, click the nearby facility button
2. You can see the facility list that around the event region
3. If you want to see the detail, click the facility that you want

03 Main Functions

## UI & Scenarios

For an app that enables everyone to enjoy sports comfortably

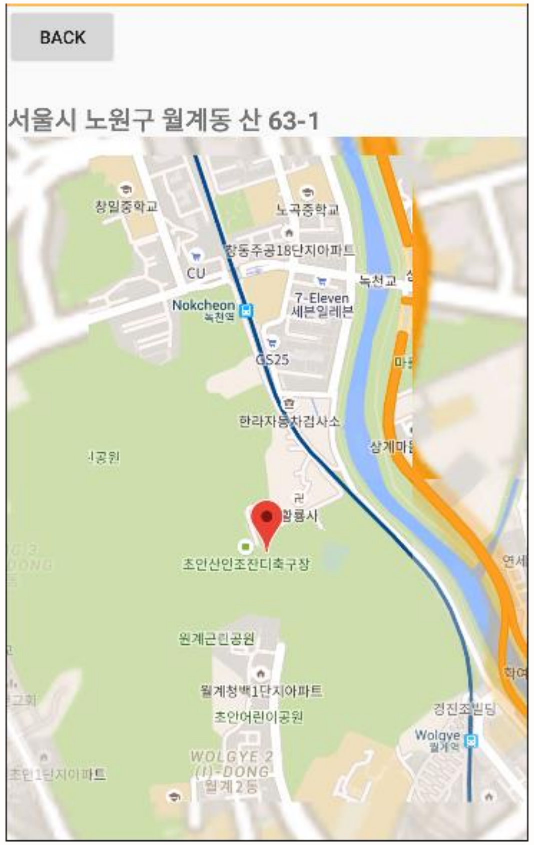
### 5. Nearby Facility



Sports Go

초인산 인조잔디 축구장(월계구장)

| | |
|---|---|
| Address | 서울시 노원구 월계동 산 63-1 |
| Capacity | 경기장 규격: 93 x 60 (11 vs 11 추천) |
| Info | 인조잔디<br>조명 있음<br>주차장 있음 |
| Link | http://gongdan.nowon.kr/rent/rent_plan_list.php?comcd=NOWON_GUMIN01 |



1. After click the facility, you can see the detail information

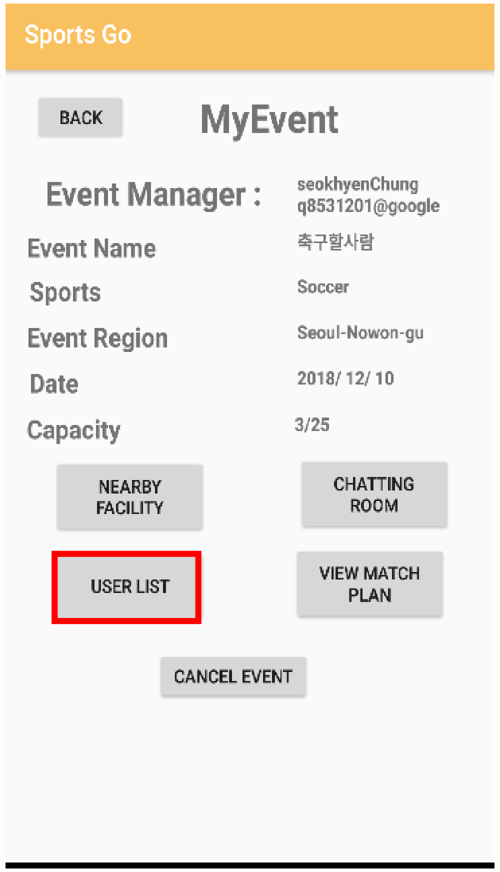
2. If you want to location of facility, click the map

3. You can see the marker as location on the map

03 Main Functions

## UI & Scenarios

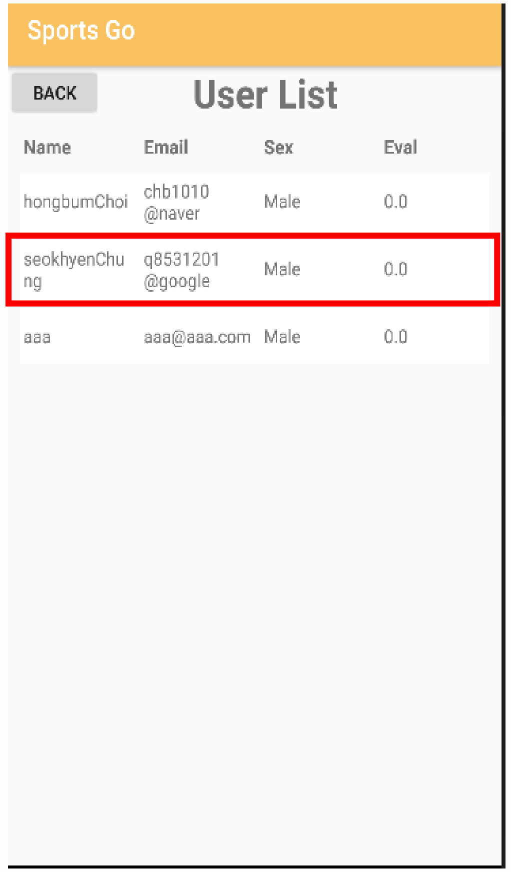
### 6. User List



Sports Go

BACK MyEvent

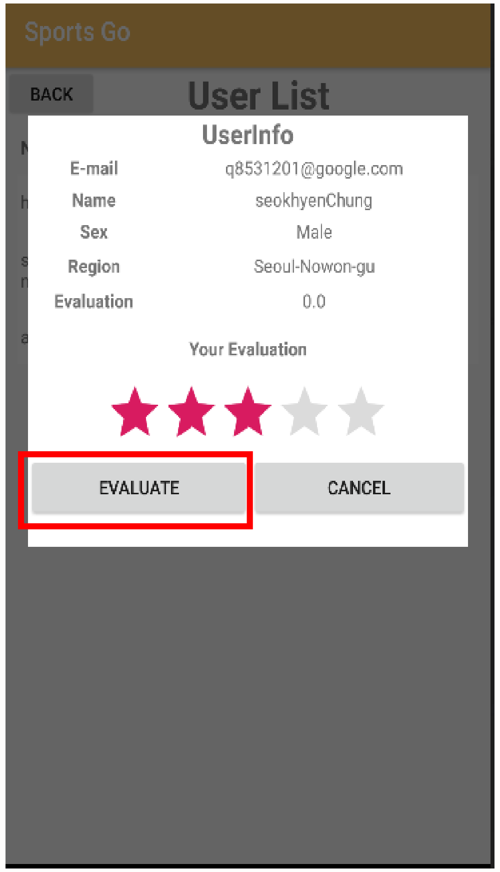
| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 3/25 |



Sports Go

BACK User List

| Name | Email | Sex | Eval |
|---|---|---|---|
| hongbumChoi | chb1010 @naver | Male | 0.0 |
| seokhyenChung | q8531201 @google | Male | 0.0 |
| aaa | aaa@aaa.com | Male | 0.0 |



Sports Go

BACK User List

UserInfo

| | |
|---|---|
| E-mail | q8531201@google.com |
| Name | seokhyenChung |
| Sex | Male |
| Region | Seoul-Nowon-gu |
| Evaluation | 0.0 |

Your Evaluation

★★★☆☆

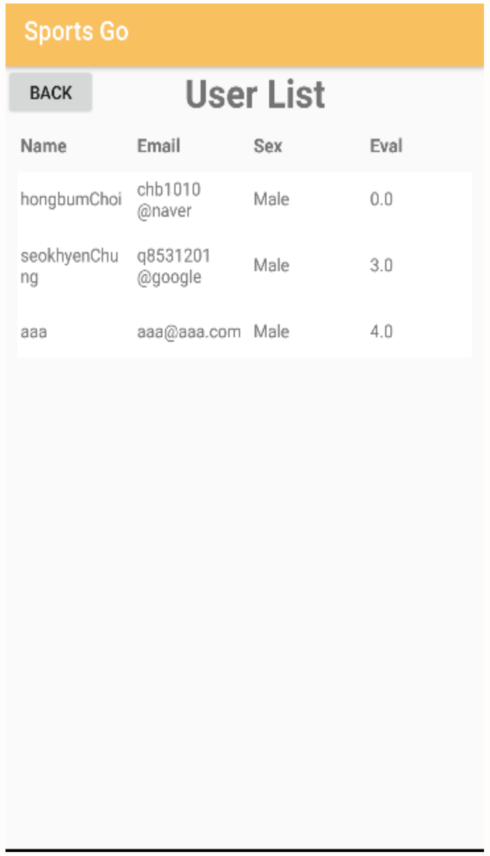
1. If you want to see the member, click user list button

2. You can see the members, if you want to see the detail, click the user

3. You can see the user information and you can evaluate members with stars

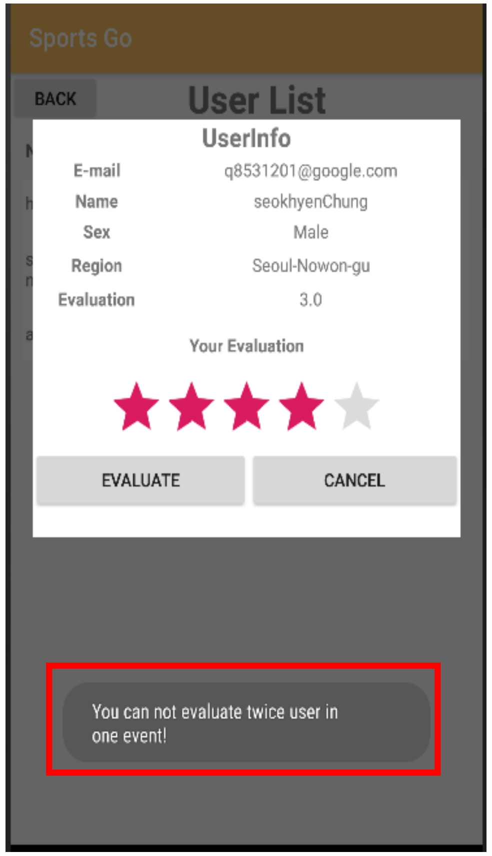
## UI & Scenarios

### 6. User List


Sports Go

BACK User List

| Name | Email | Sex | Eval |
|---|---|---|---|
| hongbumChoi | chb1010 @naver | Male | 0.0 |
| seokhyenChung | q8531201 @google | Male | 3.0 |
| aaa | aaa@aaa.com | Male | 4.0 |


Sports Go

BACK User List

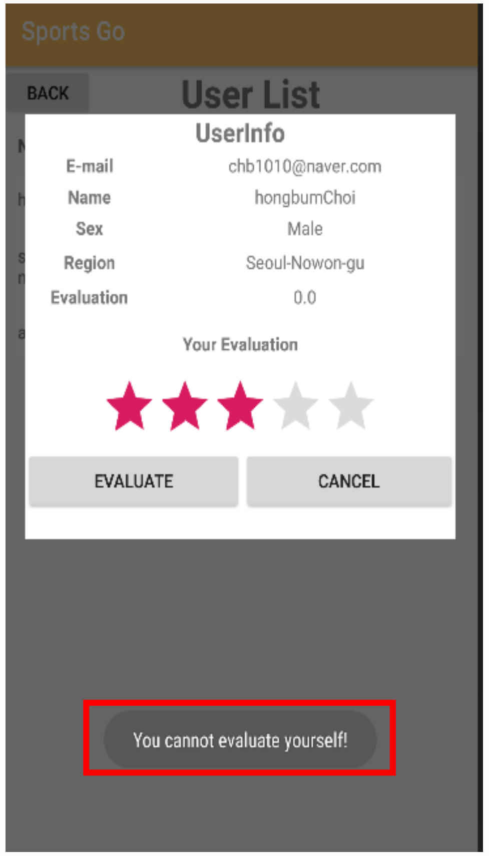
UserInfo

| | |
|---|---|
| E-mail | q8531201@google.com |
| Name | seokhyenChung |
| Sex | Male |
| Region | Seoul-Nowon-gu |
| Evaluation | 3.0 |

Your Evaluation

EVALUATE CANCEL

You can not evaluate twice user in one event!


Sports Go

BACK User List

UserInfo

| | |
|---|---|
| E-mail | chb1010@naver.com |
| Name | hongbumChoi |
| Sex | Male |
| Region | Seoul-Nowon-gu |
| Evaluation | 0.0 |

Your Evaluation

EVALUATE CANCEL

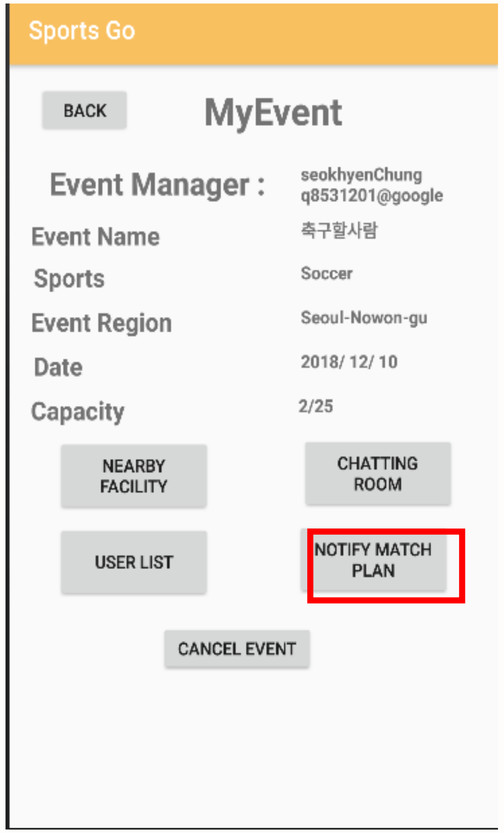
You cannot evaluate yourself!

1. When you evaluate members you can not evaluate same member twice at on event

2. Also, you can not evaluate yourself

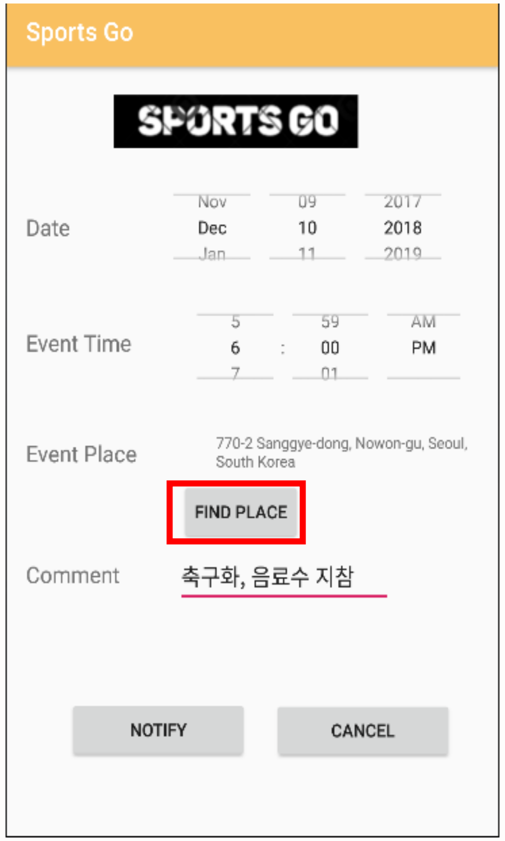
## UI & Scenarios

### 7. Notify Match Plan

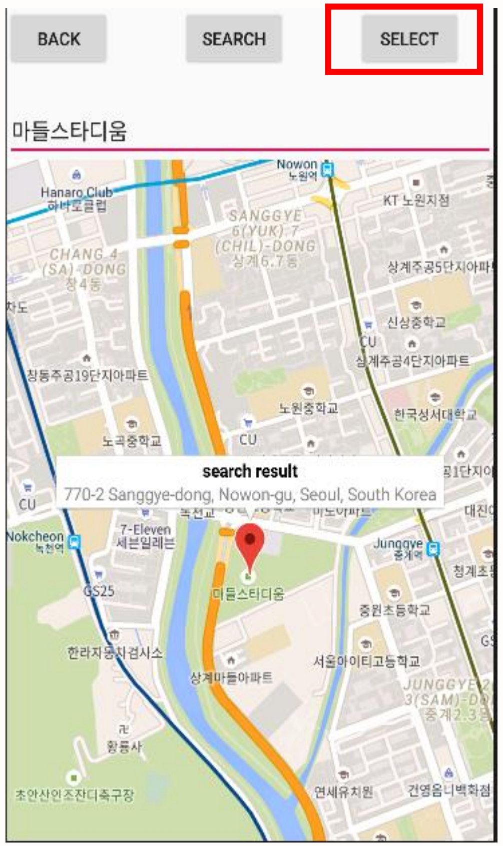


MyEvent

| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 2/25 |



| | |
|---|---|
| Date | Dec 10 2018 |
| Event Time | 6 : 00 PM |
| Event Place | 770-2 Sanggye-dong, Nowon-gu, Seoul, South Korea |
| Comment | 축구화, 음료수 지참 |



770-2 Sanggye-dong, Nowon-gu, Seoul, South Korea

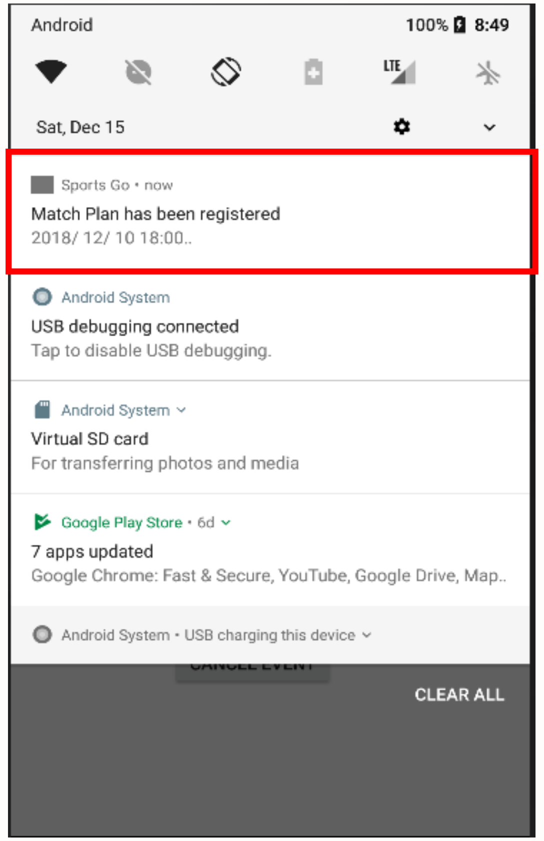
1. If match plan is made up, event manager can notify the match plan

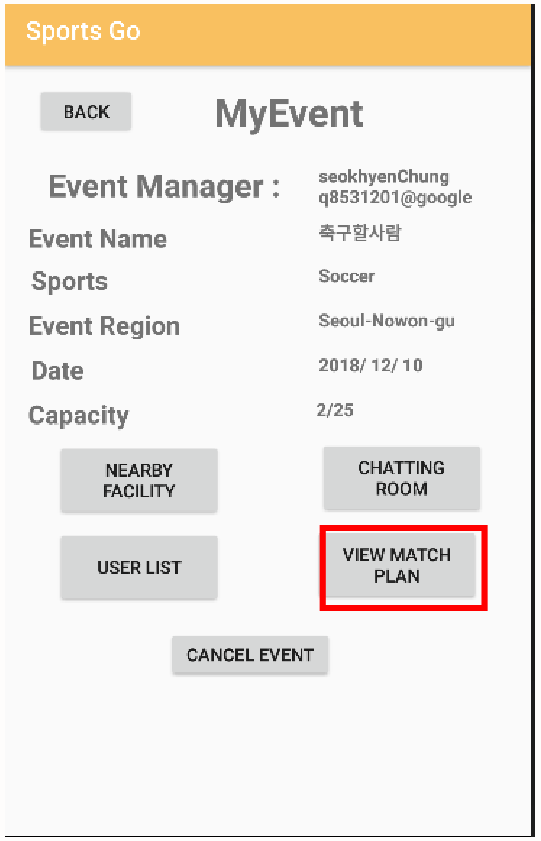
2. Fill out the form and select the event place with find place button

3. You can find place with searching the place name and click select button to set the place

## UI & Scenarios

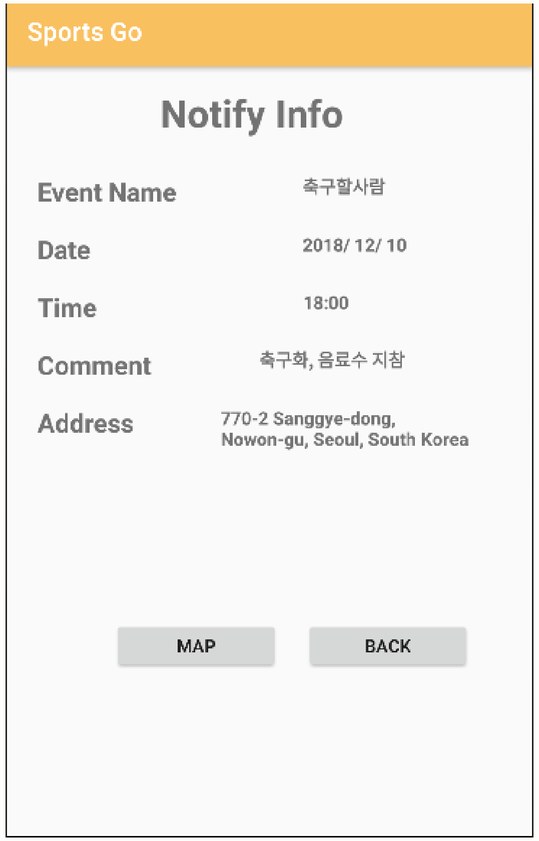
### 7. Notify Match Plan


Android 100% 8:49
Sat, Dec 15
Sports Go • now
Match Plan has been registered
2018/ 12/ 10 18:00..
Android System
USB debugging connected
Tap to disable USB debugging.
Android System
Virtual SD card
For transferring photos and media
Google Play Store • 6d
7 apps updated
Google Chrome: Fast & Secure, YouTube, Google Drive, Map..
Android System • USB charging this device



MyEvent

| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 2/25 |



Notify Info

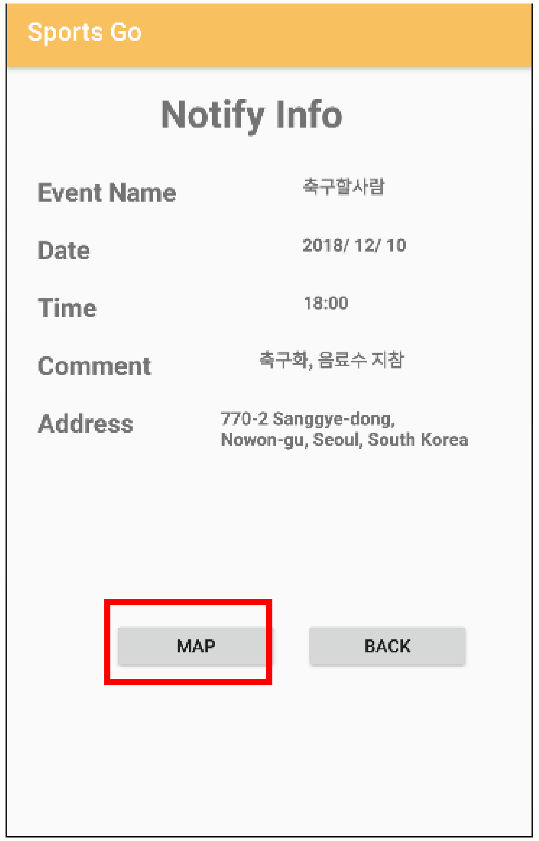
| | |
|---|---|
| Event Name | 축구할사람 |
| Date | 2018/ 12/ 10 |
| Time | 18:00 |
| Comment | 축구화, 음료수 지참 |
| Address | 770-2 Sanggye-dong, Nowon-gu, Seoul, South Korea |

1. After manager notify the plan, other members can see the notification
2. If you click the notification, you can see the match plan information

03 Main Functions

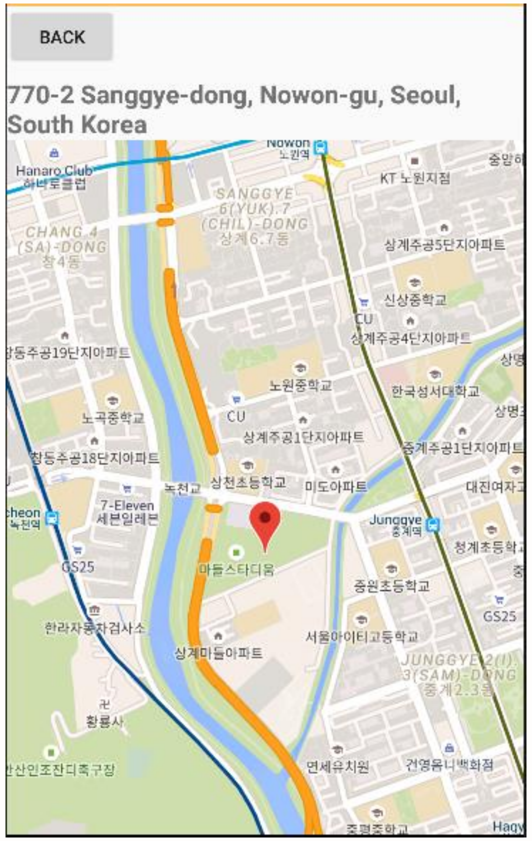
## UI & Scenarios

### 7. Notify Match Plan


Sports Go

Notify Info

| | |
|---|---|
| Event Name | 축구할사람 |
| Date | 2018/ 12/ 10 |
| Time | 18:00 |
| Comment | 축구화, 음료수 지참 |
| Address | 770-2 Sanggye-dong, Nowon-gu, Seoul, South Korea |



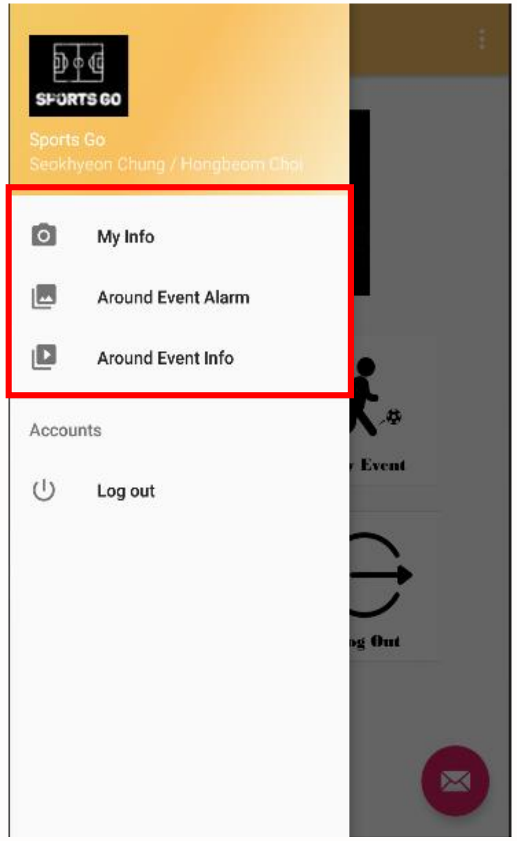
770-2 Sanggye-dong, Nowon-gu, Seoul, South Korea

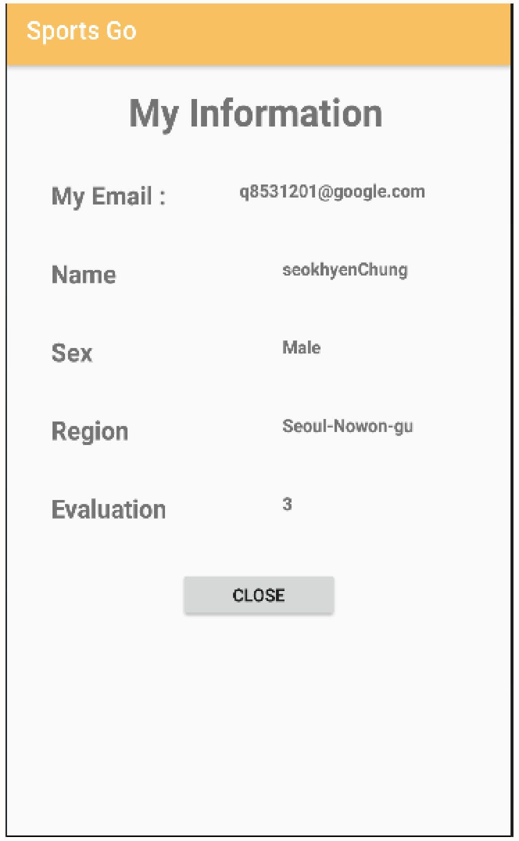
1. If you want to the location on the map, click the map button

2. You can see the marker (location) on the map

## UI & Scenarios

### 8. Navigation Drawer

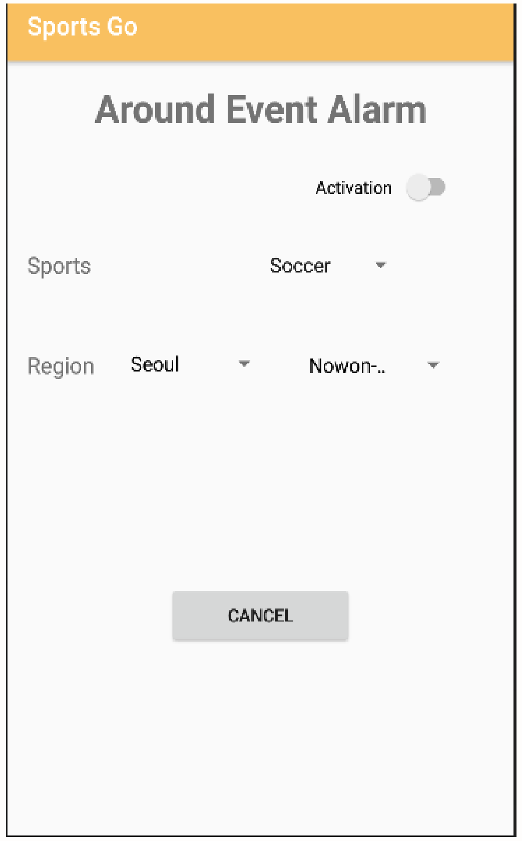



Sports Go

My Information

| | |
|---|---|
| My Email : | q8531201@google.com |
| Name | seokhyenChung |
| Sex | Male |
| Region | Seoul-Nowon-gu |
| Evaluation | 3 |

CLOSE


Sports Go

Around Event Alarm

Activation

Sports: Soccer

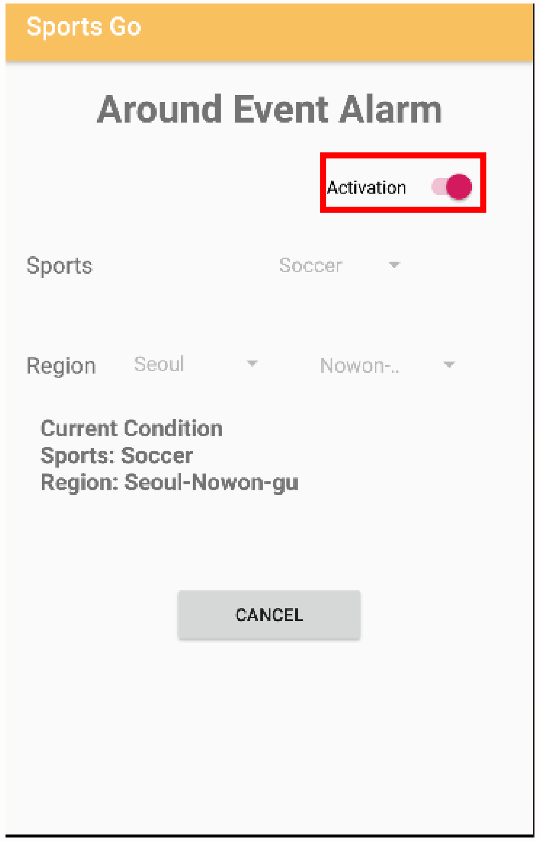
Region: Seoul, Nowon-..

CANCEL

1. At the main page, you can see the navigation drawer
2. At the navigation drawer you can see your information and you can set the around event alarm

## UI & Scenarios

### 8. Navigation Drawer


Sports Go

Around Event Alarm

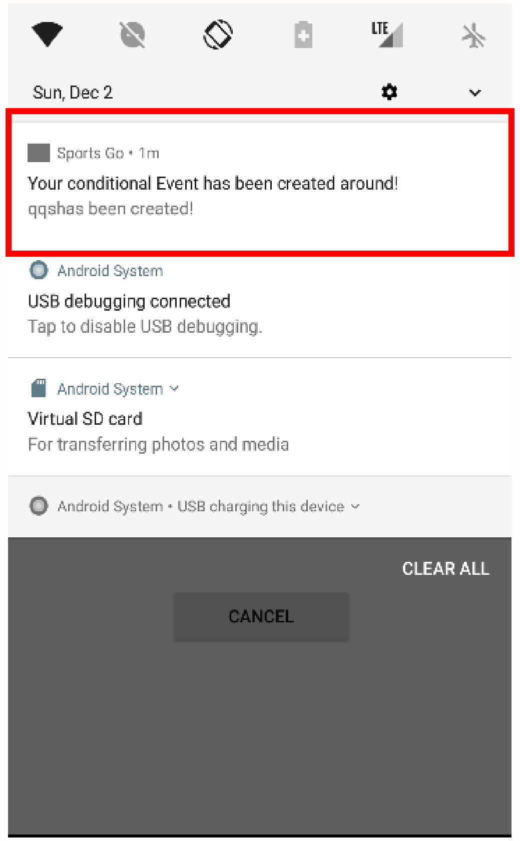
Activation

Sports Soccer

Region Seoul Nowon-..

Current Condition
Sports: Soccer
Region: Seoul-Nowon-gu

CANCEL


Sun, Dec 2

Sports Go • 1m
Your conditional Event has been created around!
qqshas been created!

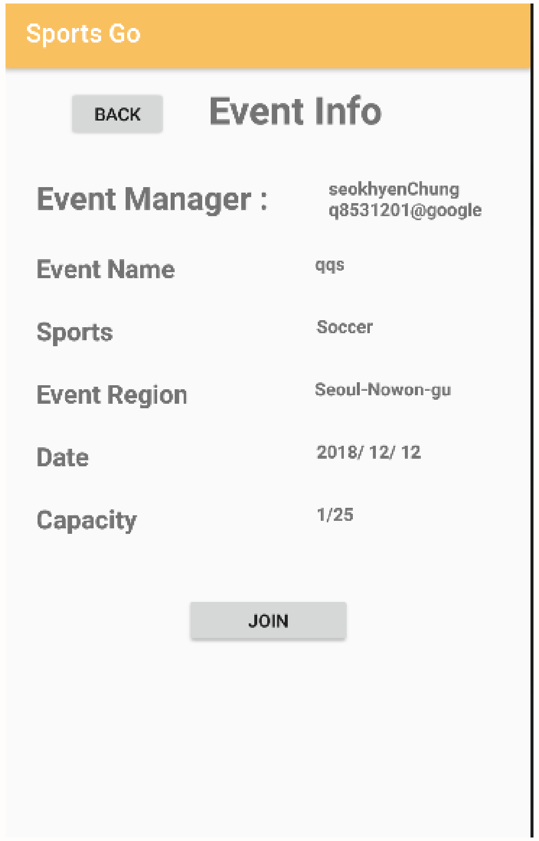
Android System
USB debugging connected
Tap to disable USB debugging.

Android System
Virtual SD card
For transferring photos and media

Android System • USB charging this device

CLEAR ALL

CANCEL


Sports Go

BACK Event Info

| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | qqs |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 12 |
| Capacity | 1/25 |

JOIN

1. You can activate around event alarm with your own conditions
2. If event that matched your condition is created, application notify the event.
3. If you click the notification, you can see the event information

## Source Codes – Github address

All source codes are here

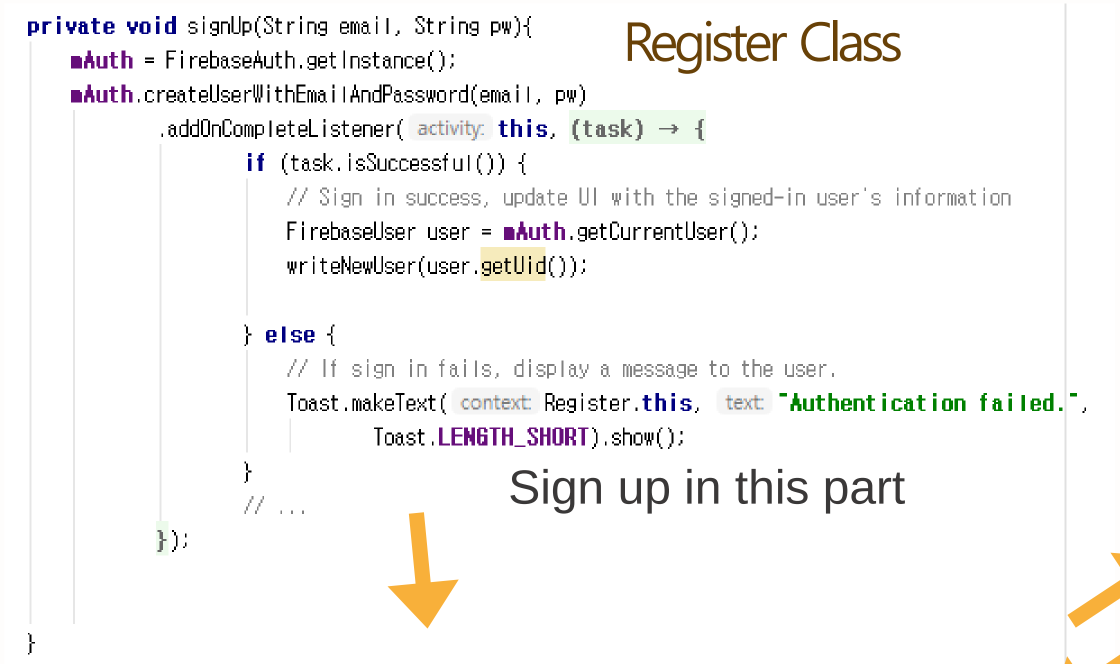
**https://github.com/Hongbeom/Sports_Go_Android_application**

03 Main Functions – Source code

## SIGN IN/UP with Firebase Authentication

### 1. Sign Up

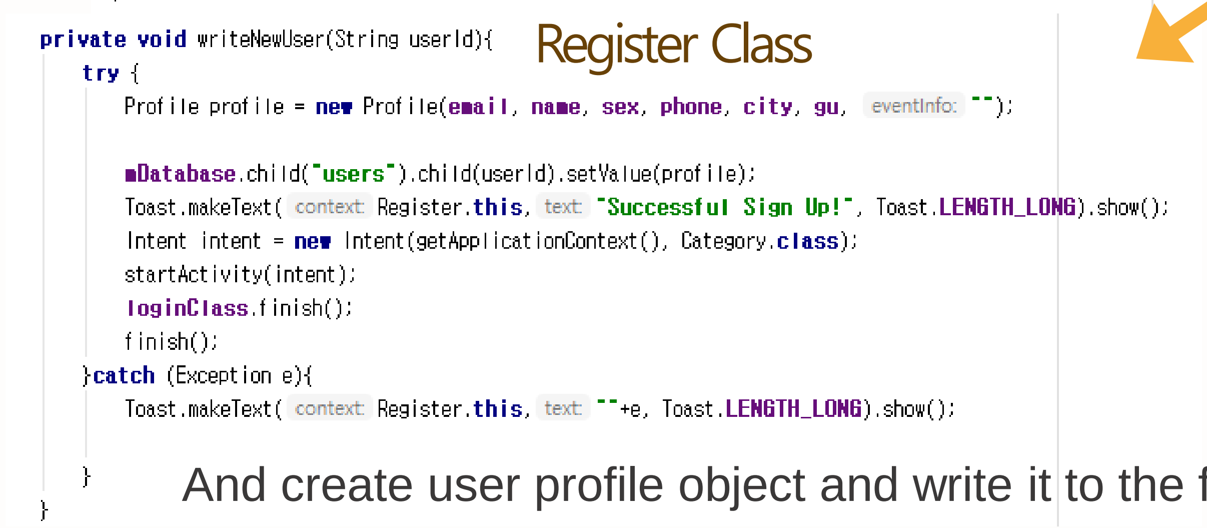
***We use Firebase Authentication and Realtime database!.***



Register Class

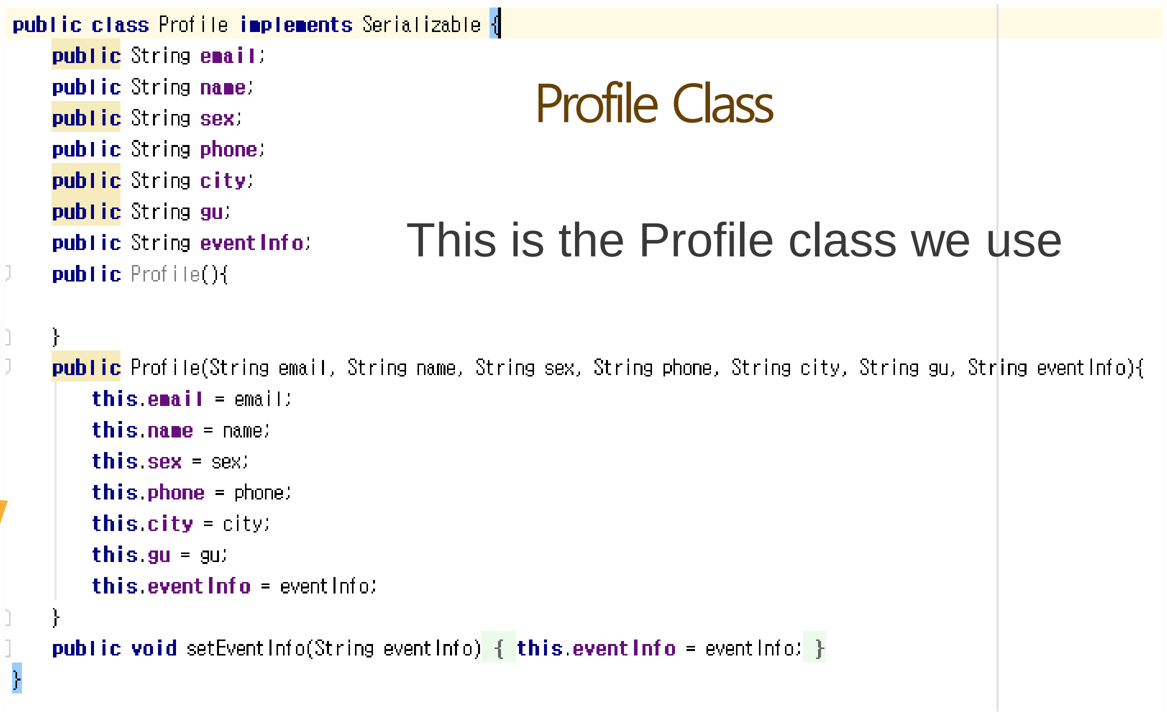
```java
private void signUp(String email, String pw){
    mAuth = FirebaseAuth.getInstance();
    mAuth.createUserWithEmailAndPassword(email, pw)
            .addOnCompleteListener( activity: this, (task) → {
                if (task.isSuccessful()) {
                    // Sign in success, update UI with the signed-in user's information
                    FirebaseUser user = mAuth.getCurrentUser();
                    writeNewUser(user.getUid());

                } else {
                    // If sign in fails, display a message to the user.
                    Toast.makeText( context: Register.this,  text: "Authentication failed.",
                            Toast.LENGTH_SHORT).show();
                }
                // ...
            });

}
```

Sign up in this part



Register Class

```java
private void writeNewUser(String userId){
    try {
        Profile profile = new Profile(email, name, sex, phone, city, gu,  eventInfo: "");

        mDatabase.child("users").child(userId).setValue(profile);
        Toast.makeText( context: Register.this,  text: "Successful Sign Up!", Toast.LENGTH_LONG).show();
        Intent intent = new Intent(getApplicationContext(), Category.class);
        startActivity(intent);
        loginClass.finish();
        finish();
    }catch (Exception e){
        Toast.makeText( context: Register.this,  text: ""+e, Toast.LENGTH_LONG).show();

    }
}
```



Profile Class

```java
public class Profile implements Serializable {
    public String email;
    public String name;
    public String sex;
    public String phone;
    public String city;
    public String gu;
    public String eventInfo;
    public Profile(){

    }
    public Profile(String email, String name, String sex, String phone, String city, String gu, String eventInfo){
        this.email = email;
        this.name = name;
        this.sex = sex;
        this.phone = phone;
        this.city = city;
        this.gu = gu;
        this.eventInfo = eventInfo;
    }
    public void setEventInfo(String eventInfo) { this.eventInfo = eventInfo; }
}
```

This is the Profile class we use

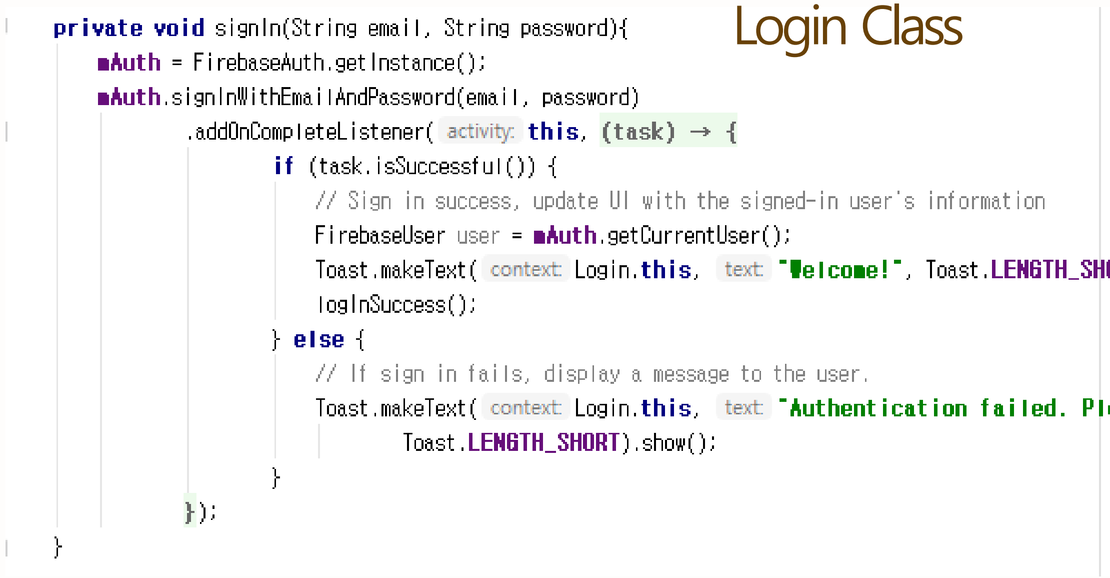
And create user profile object and write it to the firebase database

03 Main Functions – Source code

## SIGN IN/UP with Firebase Authentication

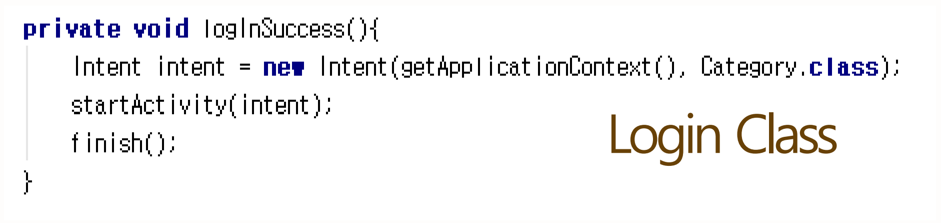
### 2. Sign In

Sign up in this part



Login Class

```
private void signIn(String email, String password){
    mAuth = FirebaseAuth.getInstance();
    mAuth.signInWithEmailAndPassword(email, password)
            .addOnCompleteListener( activity: this, (task) -> {
                if (task.isSuccessful()) {
                    // Sign in success, update UI with the signed-in user's information
                    FirebaseUser user = mAuth.getCurrentUser();
                    Toast.makeText( context: Login.this,  text: "Welcome!", Toast.LENGTH_SHORT).show();
                    logInSuccess();
                } else {
                    // If sign in fails, display a message to the user.
                    Toast.makeText( context: Login.this,  text: "Authentication failed. Please check your account.",
                            Toast.LENGTH_SHORT).show();
                }
            });
}
```



```
private void logInSuccess(){
    Intent intent = new Intent(getApplicationContext(), Category.class);
    startActivity(intent);
    finish();
}
```

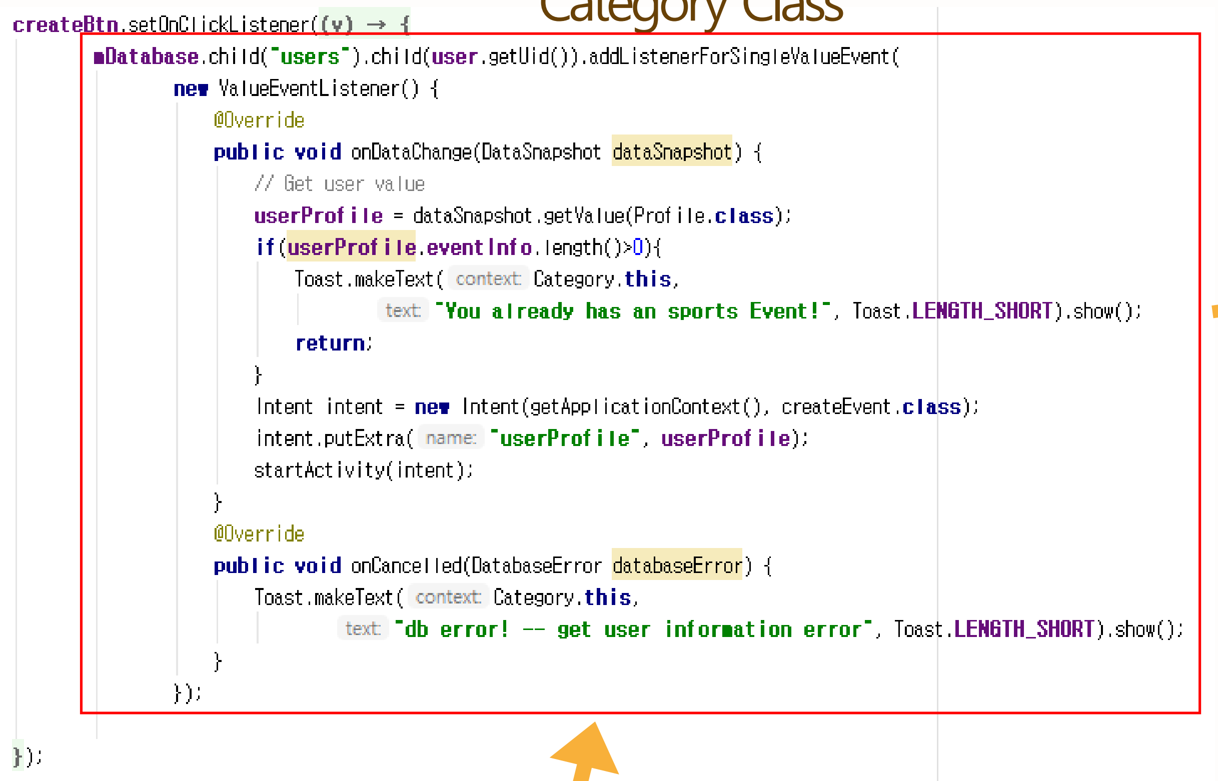
Login Class

03 Main Functions – Source code

## Sports Event Handling

### 3. Create Event

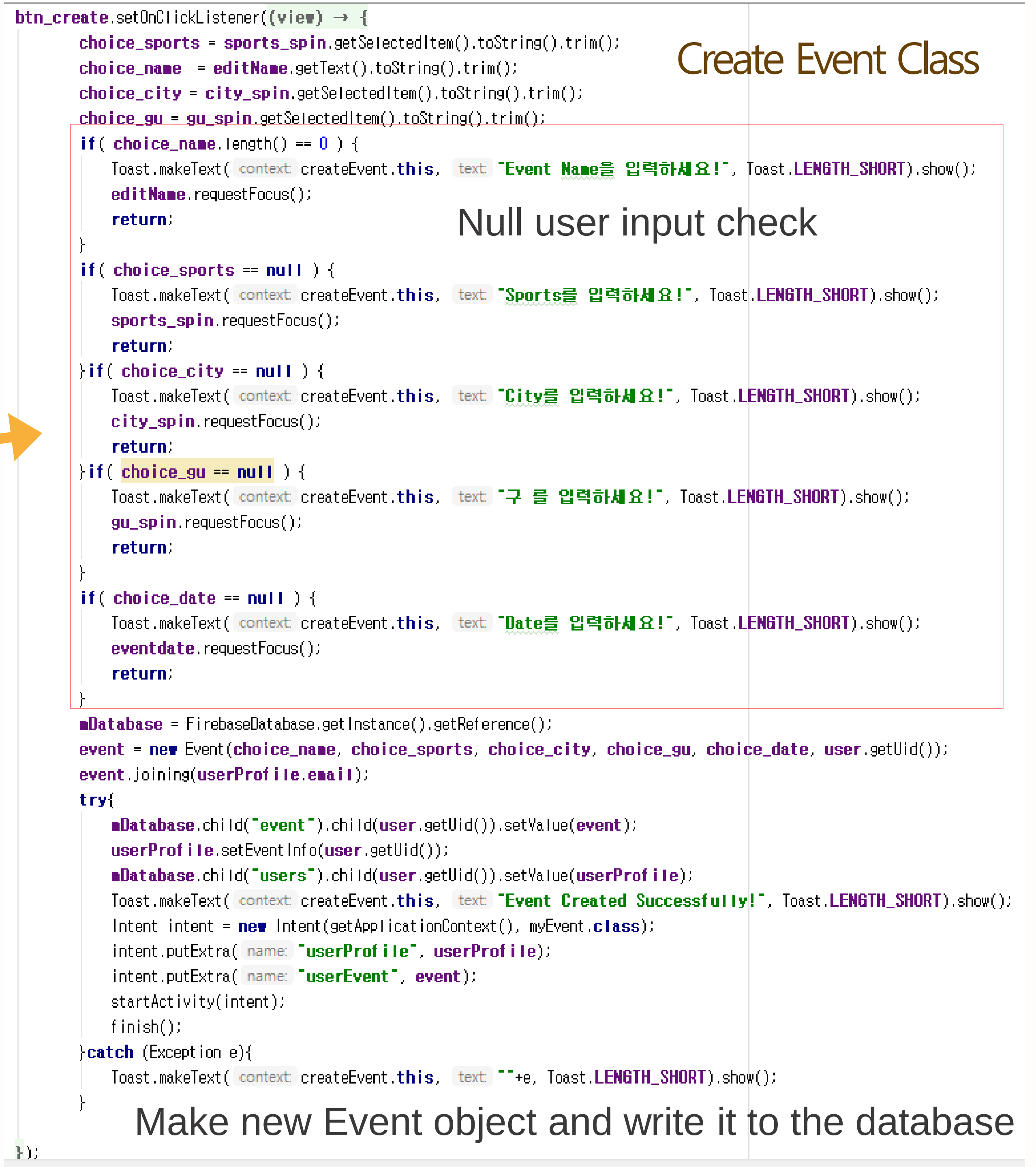
Category Class



```
createBtn.setOnClickListener((v) → {
    mDatabase.child("users").child(user.getUid()).addListenerForSingleValueEvent(
            new ValueEventListener() {
                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {
                    // Get user value
                    userProfile = dataSnapshot.getValue(Profile.class);
                    if(userProfile.eventInfo.length()>0){
                        Toast.makeText( context: Category.this,
                                 text: "You already has an sports Event!", Toast.LENGTH_SHORT).show();
                        return;
                    }
                    Intent intent = new Intent(getApplicationContext(), createEvent.class);
                    intent.putExtra( name: "userProfile", userProfile);
                    startActivity(intent);
                }
                @Override
                public void onCancelled(DatabaseError databaseError) {
                    Toast.makeText( context: Category.this,
                            text: "db error! -- get user information error", Toast.LENGTH_SHORT).show();
                }
            });

});
```

Search the user profile from firebase database and start the activity that create event with the user profile

Create Event Class



```
btn_create.setOnClickListener((view) → {
    choice_sports = sports_spin.getSelectedItem().toString().trim();
    choice_name  = editName.getText().toString().trim();
    choice_city = city_spin.getSelectedItem().toString().trim();
    choice_gu = gu_spin.getSelectedItem().toString().trim();
    if( choice_name.length() == 0 ) {
        Toast.makeText( context: createEvent.this,  text: "Event Name을 입력하세요!", Toast.LENGTH_SHORT).show();
        editName.requestFocus();
        return;
    }
    if( choice_sports == null ) {
        Toast.makeText( context: createEvent.this,  text: "Sports를 입력하세요!", Toast.LENGTH_SHORT).show();
        sports_spin.requestFocus();
        return;
    }if( choice_city == null ) {
        Toast.makeText( context: createEvent.this,  text: "City를 입력하세요!", Toast.LENGTH_SHORT).show();
        city_spin.requestFocus();
        return;
    }if( choice_gu == null ) {
        Toast.makeText( context: createEvent.this,  text: "구 를 입력하세요!", Toast.LENGTH_SHORT).show();
        gu_spin.requestFocus();
        return;
    }
    if( choice_date == null ) {
        Toast.makeText( context: createEvent.this,  text: "Date를 입력하세요!", Toast.LENGTH_SHORT).show();
        eventdate.requestFocus();
        return;
    }
    mDatabase = FirebaseDatabase.getInstance().getReference();
    event = new Event(choice_name, choice_sports, choice_city, choice_gu, choice_date, user.getUid());
    event.joining(userProfile.email);
    try{
        mDatabase.child("event").child(user.getUid()).setValue(event);
        userProfile.setEventInfo(user.getUid());
        mDatabase.child("users").child(user.getUid()).setValue(userProfile);
        Toast.makeText( context: createEvent.this,  text: "Event Created Successfully!", Toast.LENGTH_SHORT).show();
        Intent intent = new Intent(getApplicationContext(), myEvent.class);
        intent.putExtra( name: "userProfile", userProfile);
        intent.putExtra( name: "userEvent", event);
        startActivity(intent);
        finish();
    }catch (Exception e){
        Toast.makeText( context: createEvent.this,  text: ""+e, Toast.LENGTH_SHORT).show();
    }
});
```

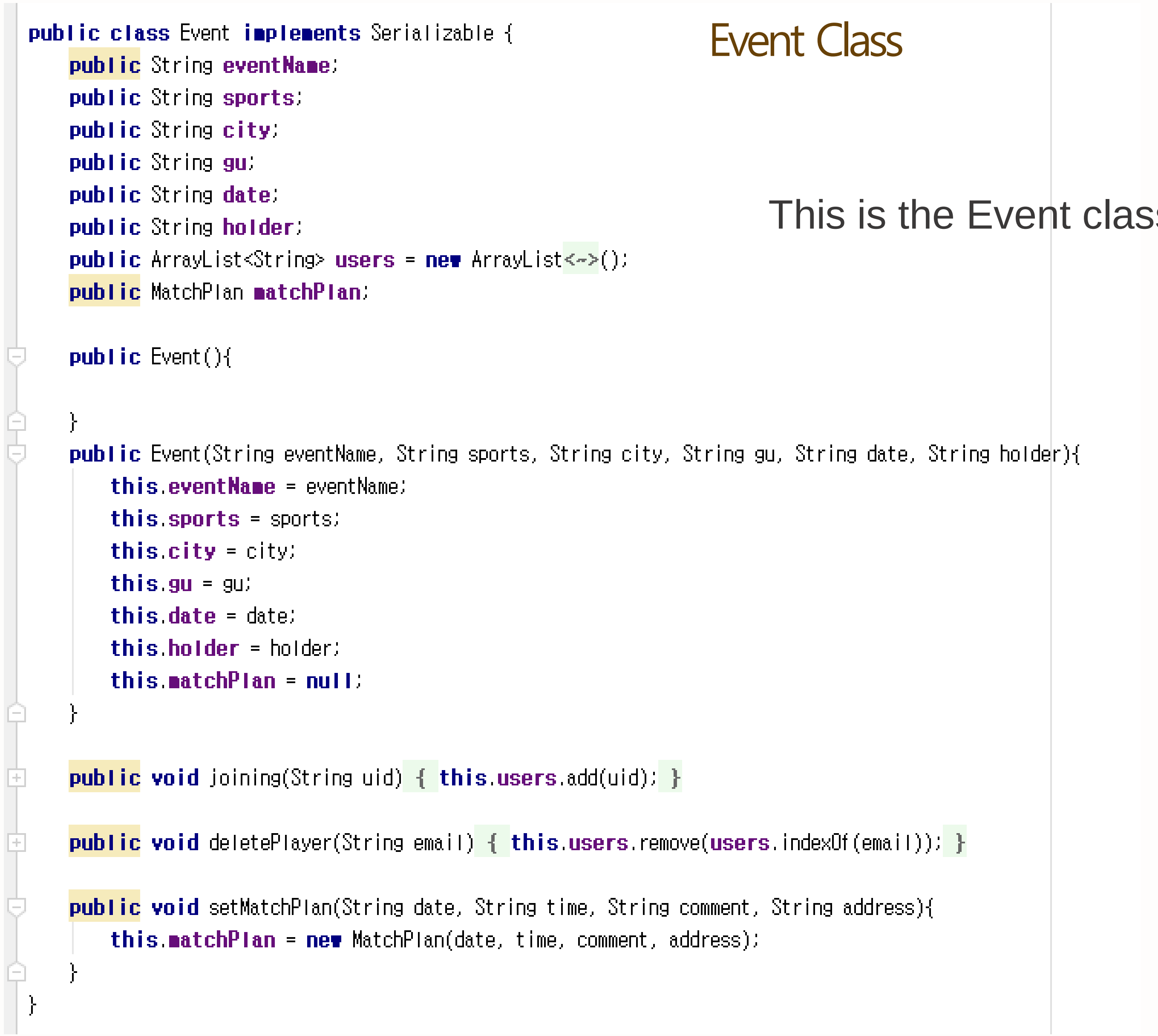
Null user input check

Make new Event object and write it to the database

03 Main Functions – Source code

## Sports Event Handling

### 3. Create Event



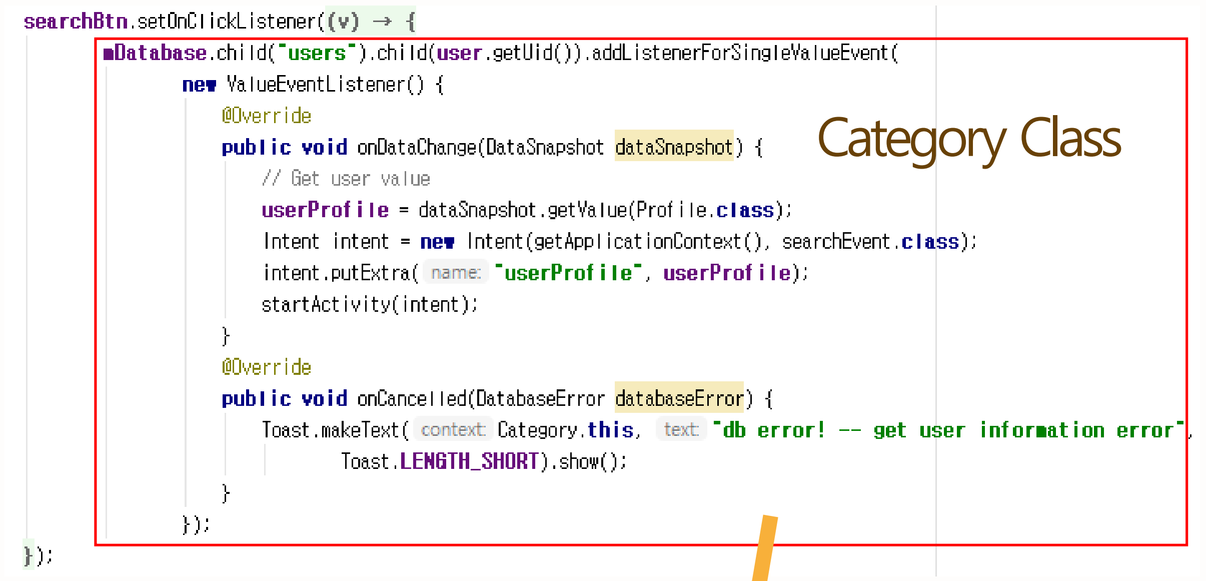
```
public class Event implements Serializable {
    public String eventName;
    public String sports;
    public String city;
    public String gu;
    public String date;
    public String holder;
    public ArrayList<String> users = new ArrayList<~>();
    public MatchPlan matchPlan;

    public Event(){

    }
    public Event(String eventName, String sports, String city, String gu, String date, String holder){
        this.eventName = eventName;
        this.sports = sports;
        this.city = city;
        this.gu = gu;
        this.date = date;
        this.holder = holder;
        this.matchPlan = null;
    }

    public void joining(String uid) { this.users.add(uid); }

    public void deletePlayer(String email) { this.users.remove(users.indexOf(email)); }

    public void setMatchPlan(String date, String time, String comment, String address){
        this.matchPlan = new MatchPlan(date, time, comment, address);
    }
}
```

Event Class

This is the Event class we use

## Sports Event Handling

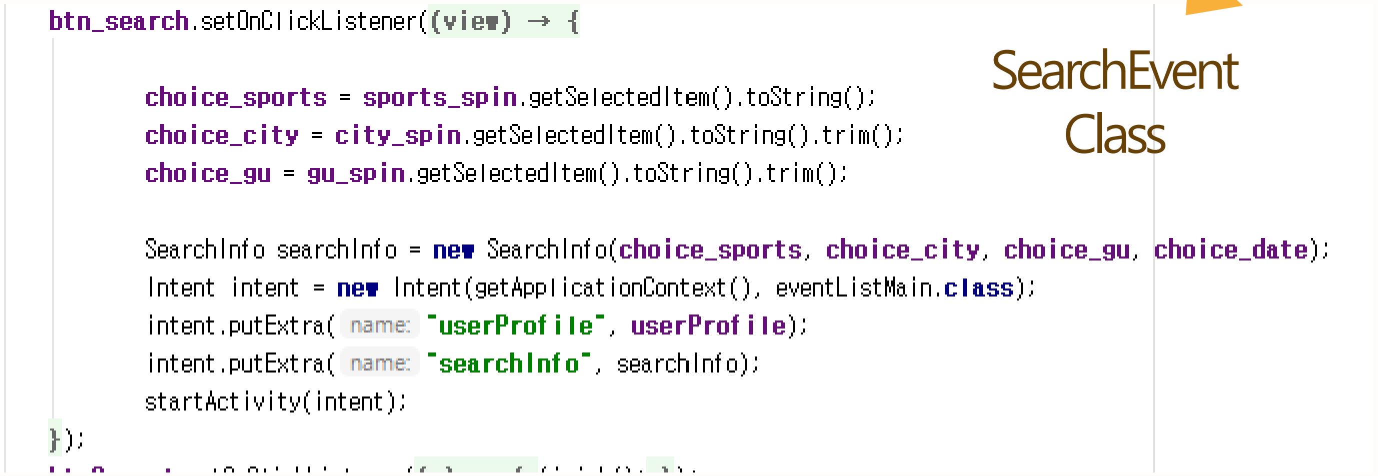
### 4. Search Event



```
searchBtn.setOnClickListener((v) → {
        mDatabase.child("users").child(user.getUid()).addListenerForSingleValueEvent(
                new ValueEventListener() {
                    @Override
                    public void onDataChange(DataSnapshot dataSnapshot) {
                        // Get user value
                        userProfile = dataSnapshot.getValue(Profile.class);
                        Intent intent = new Intent(getApplicationContext(), searchEvent.class);
                        intent.putExtra( name: "userProfile", userProfile);
                        startActivity(intent);
                    }
                    @Override
                    public void onCancelled(DatabaseError databaseError) {
                        Toast.makeText( context: Category.this,  text: "db error! -- get user information error",
                                Toast.LENGTH_SHORT).show();
                    }
                });
});
```

Category Class

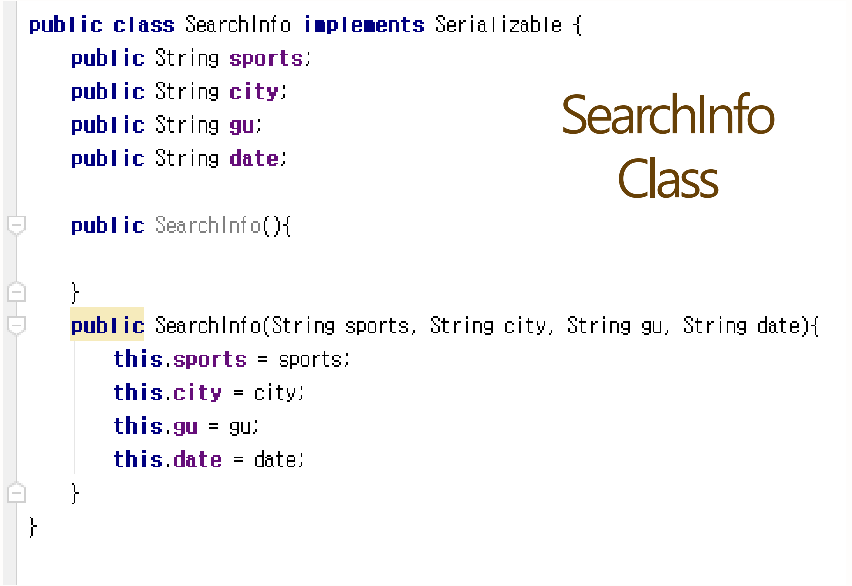
Search the user profile from firebase database and start the activity that search event with the user profile



```
btn_search.setOnClickListener((view) → {

        choice_sports = sports_spin.getSelectedItem().toString();
        choice_city = city_spin.getSelectedItem().toString().trim();
        choice_gu = gu_spin.getSelectedItem().toString().trim();

        SearchInfo searchInfo = new SearchInfo(choice_sports, choice_city, choice_gu, choice_date);
        Intent intent = new Intent(getApplicationContext(), eventListMain.class);
        intent.putExtra( name: "userProfile", userProfile);
        intent.putExtra( name: "searchInfo", searchInfo);
        startActivity(intent);
});
```

SearchEvent Class

Make the search information object from user input and start the activity eventListMain where the actual searching would be occurred



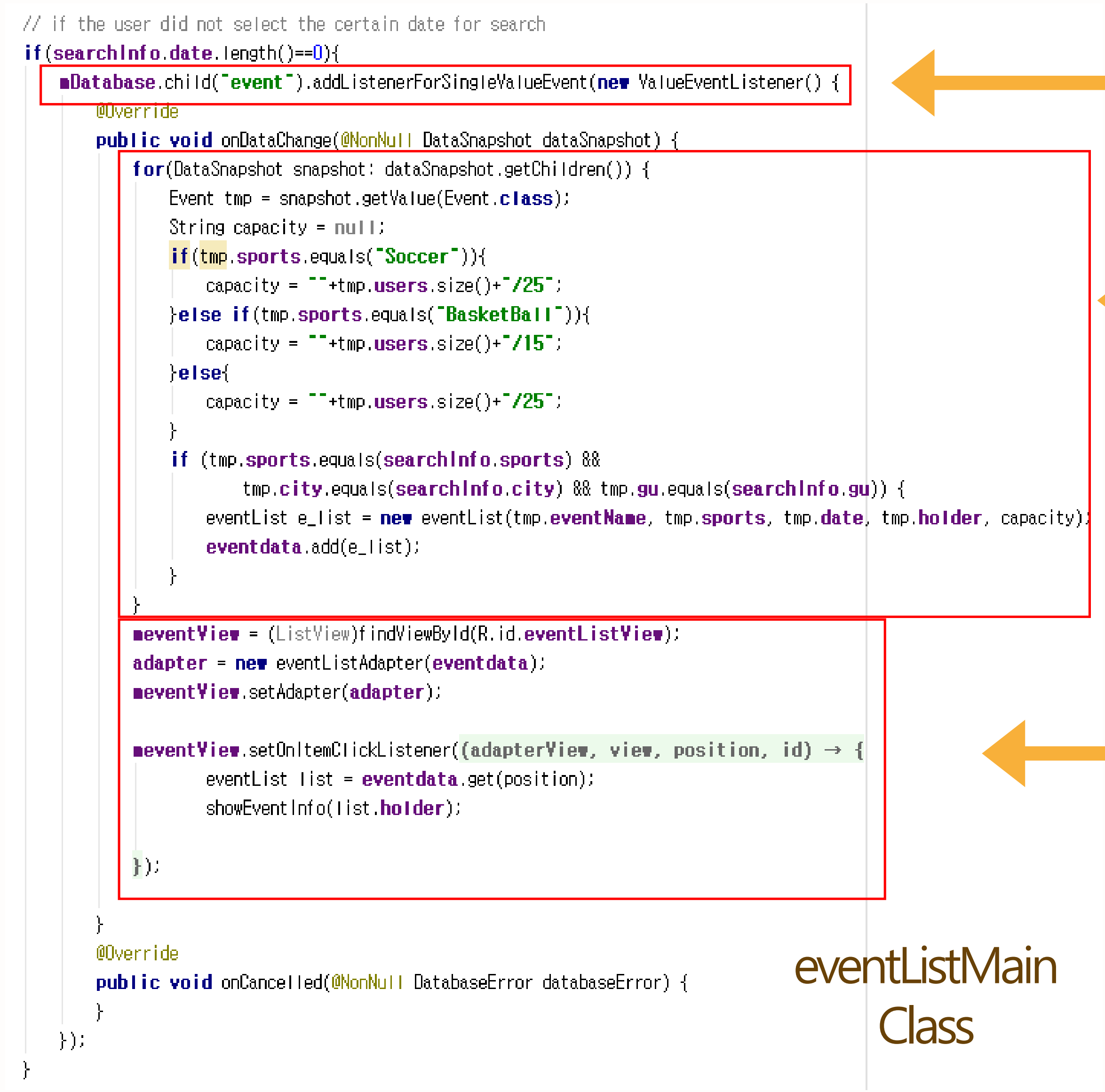
```
public class SearchInfo implements Serializable {
    public String sports;
    public String city;
    public String gu;
    public String date;

    public SearchInfo(){

    }
    public SearchInfo(String sports, String city, String gu, String date){
        this.sports = sports;
        this.city = city;
        this.gu = gu;
        this.date = date;
    }
}
```

SearchInfo Class

03 Main Functions – Source code

## Sports Event Handling

### 4. Search Event – Two cases: 1) User did not select the certain date for sports event and search



```
// if the user did not select the certain date for search
if(searchInfo.date.length()==0){
    mDatabase.child("event").addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            for(DataSnapshot snapshot: dataSnapshot.getChildren()) {
                Event tmp = snapshot.getValue(Event.class);
                String capacity = null;
                if(tmp.sports.equals("Soccer")){
                    capacity = ""+tmp.users.size()+"/25";
                }else if(tmp.sports.equals("BasketBall")){
                    capacity = ""+tmp.users.size()+"/15";
                }else{
                    capacity = ""+tmp.users.size()+"/25";
                }
                if (tmp.sports.equals(searchInfo.sports) &&
                        tmp.city.equals(searchInfo.city) && tmp.gu.equals(searchInfo.gu)) {
                    eventList e_list = new eventList(tmp.eventName, tmp.sports, tmp.date, tmp.holder, capacity);
                    eventdata.add(e_list);
                }
            }
            meventView = (ListView)findViewById(R.id.eventListView);
            adapter = new eventListAdapter(eventdata);
            meventView.setAdapter(adapter);

            meventView.setOnItemClickListener((adapterView, view, position, id) -> {
                    eventList list = eventdata.get(position);
                    showEventInfo(list.holder);

            });

        }
        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {
        }
    });
}
```

Read all events from the database

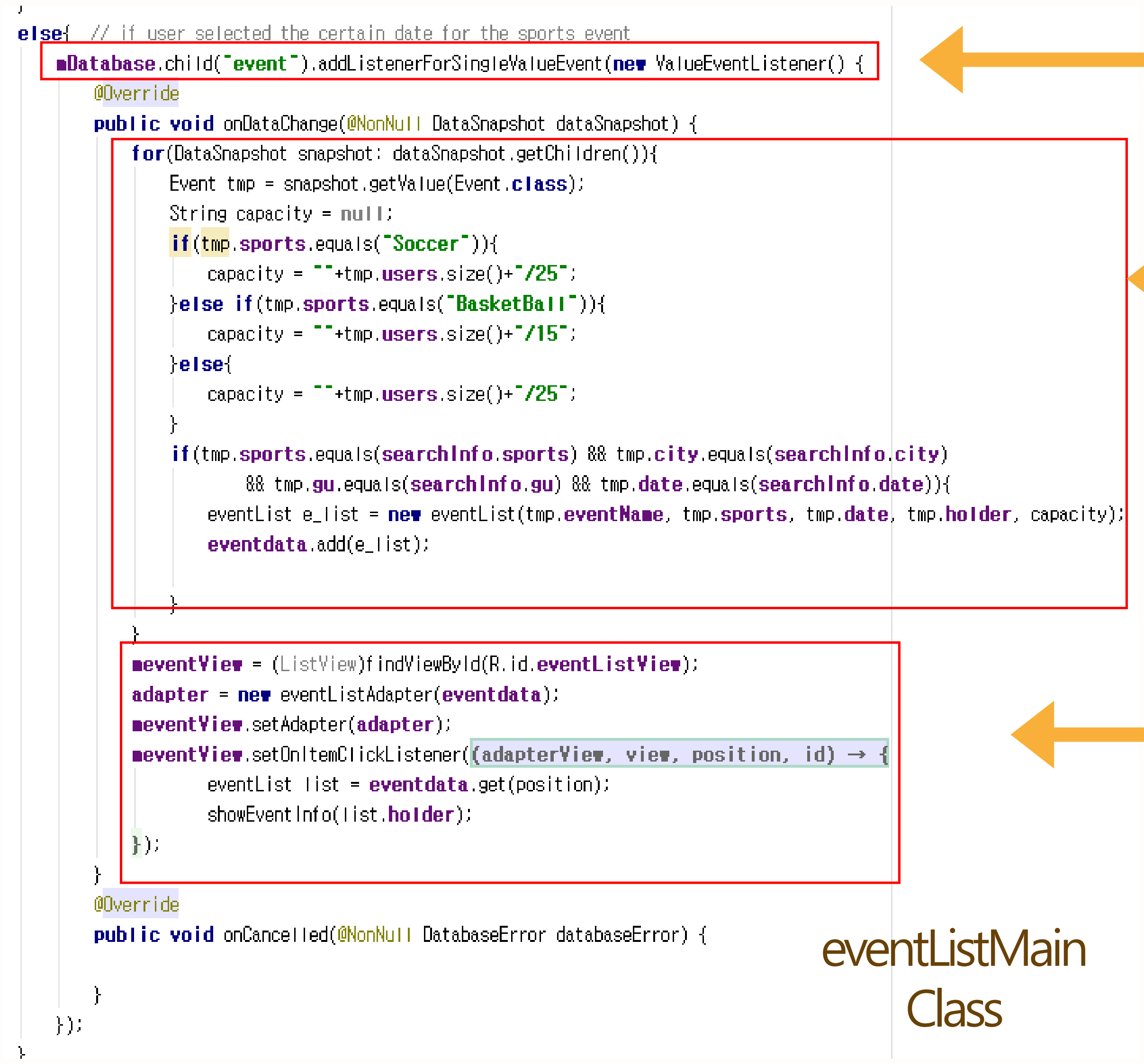
Filter each events whether the event features are same from the user search features. If the event’s features are same as the user features, add it to the array list.

After the filtering the events, make the list view in layout with the filtered events.

eventListMain Class

## Sports Event Handling

### 4. Search Event – Two cases: 2) User selected the certain date for sports event and search



```
}
else{  // if user selected the certain date for the sports event
    mDatabase.child("event").addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            for(DataSnapshot snapshot: dataSnapshot.getChildren()){
                Event tmp = snapshot.getValue(Event.class);
                String capacity = null;
                if(tmp.sports.equals("Soccer")){
                    capacity = ""+tmp.users.size()+"/25";
                }else if(tmp.sports.equals("BasketBall")){
                    capacity = ""+tmp.users.size()+"/15";
                }else{
                    capacity = ""+tmp.users.size()+"/25";
                }
                if(tmp.sports.equals(searchInfo.sports) && tmp.city.equals(searchInfo.city)
                        && tmp.gu.equals(searchInfo.gu) && tmp.date.equals(searchInfo.date)){
                    eventList e_list = new eventList(tmp.eventName, tmp.sports, tmp.date, tmp.holder, capacity);
                    eventdata.add(e_list);

                }
            }
            meventView = (ListView)findViewById(R.id.eventListView);
            adapter = new eventListAdapter(eventdata);
            meventView.setAdapter(adapter);
            meventView.setOnItemClickListener((adapterView, view, position, id) → {
                    eventList list = eventdata.get(position);
                    showEventInfo(list.holder);
            });
        }
        @Override
        public void onCancelled(@NonNull DatabaseError databaseError) {

        }
    });
}
```

Read all events from the database

Filter each events whether the event features are same from the user search features. If the event's features are same as the user features, add it to the array list.

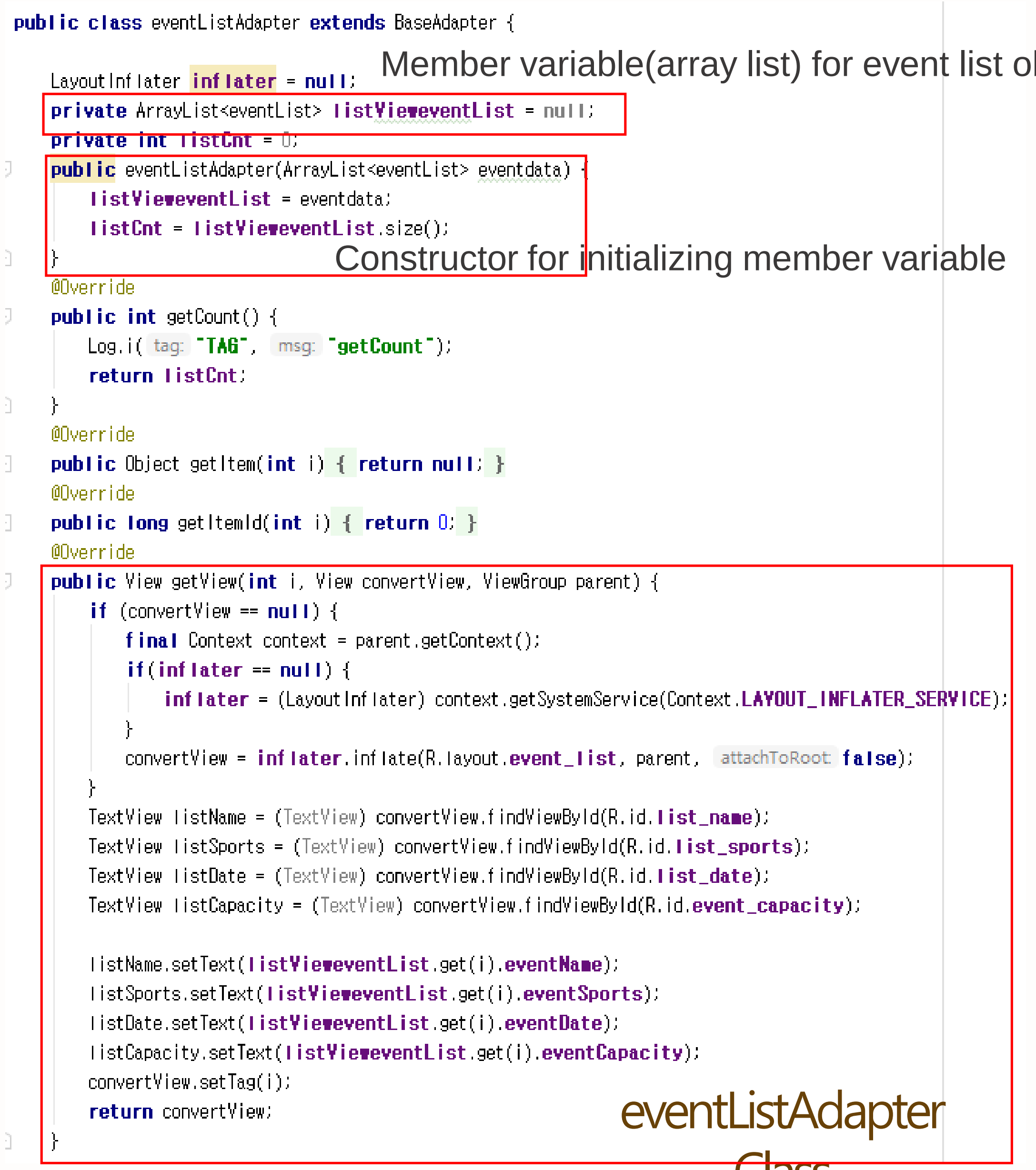
After the filtering the events, make the list view in layout with the filtered events.

eventListMain Class

03 Main Functions – Source code

## Sports Event Handling

## 4. Search Event – Adapter and List class to show the event list



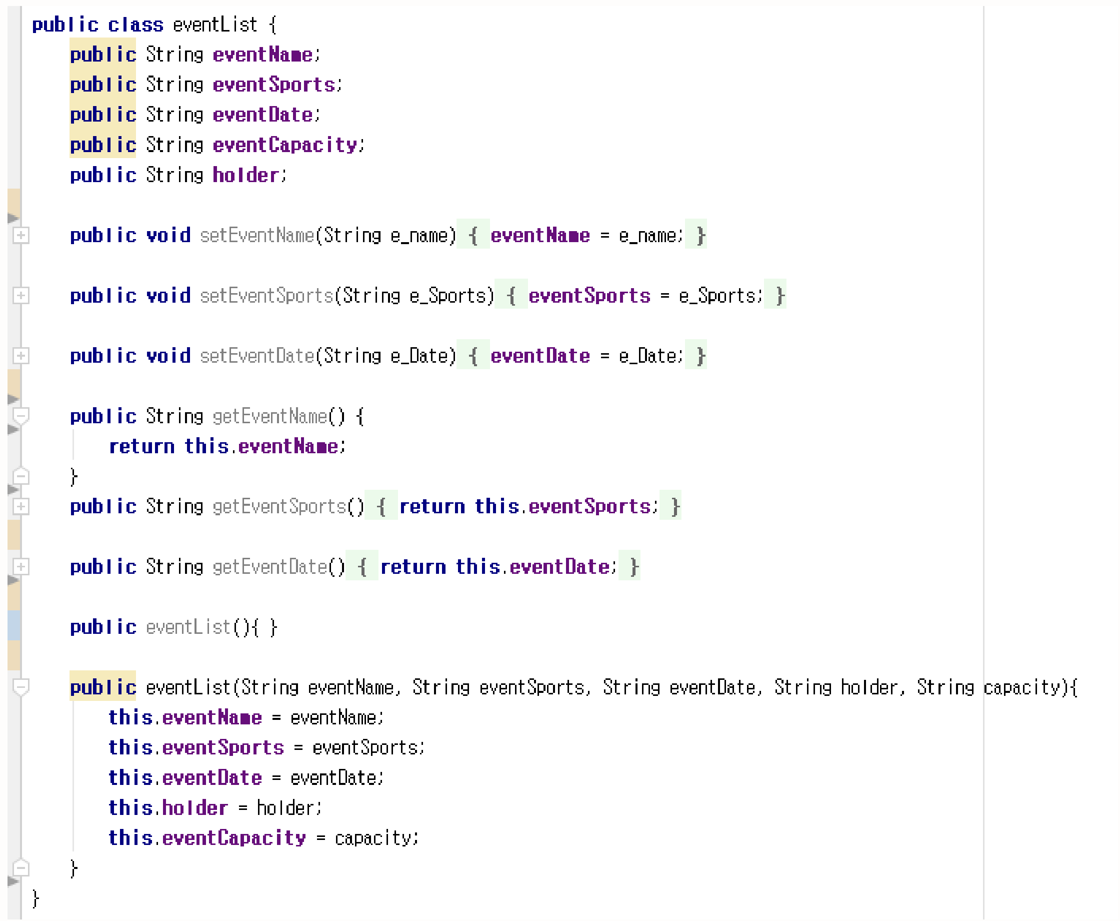
```
public class eventListAdapter extends BaseAdapter {

    LayoutInflater inflater = null;
    private ArrayList<eventList> listVieweventList = null;
    private int listCnt = 0;
    public eventListAdapter(ArrayList<eventList> eventdata) {
        listVieweventList = eventdata;
        listCnt = listVieweventList.size();
    }
    @Override
    public int getCount() {
        Log.i( tag: "TAG", msg: "getCount");
        return listCnt;
    }
    @Override
    public Object getItem(int i) { return null; }
    @Override
    public long getItemId(int i) { return 0; }
    @Override
    public View getView(int i, View convertView, ViewGroup parent) {
        if (convertView == null) {
            final Context context = parent.getContext();
            if(inflater == null) {
                inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
            }
            convertView = inflater.inflate(R.layout.event_list, parent, attachToRoot: false);
        }
        TextView listName = (TextView) convertView.findViewById(R.id.list_name);
        TextView listSports = (TextView) convertView.findViewById(R.id.list_sports);
        TextView listDate = (TextView) convertView.findViewById(R.id.list_date);
        TextView listCapacity = (TextView) convertView.findViewById(R.id.event_capacity);

        listName.setText(listVieweventList.get(i).eventName);
        listSports.setText(listVieweventList.get(i).eventSports);
        listDate.setText(listVieweventList.get(i).eventDate);
        listCapacity.setText(listVieweventList.get(i).eventCapacity);
        convertView.setTag(i);
        return convertView;
    }
```

Member variable(array list) for event list object

Constructor for initializing member variable

eventListAdapter Class

eventList Class

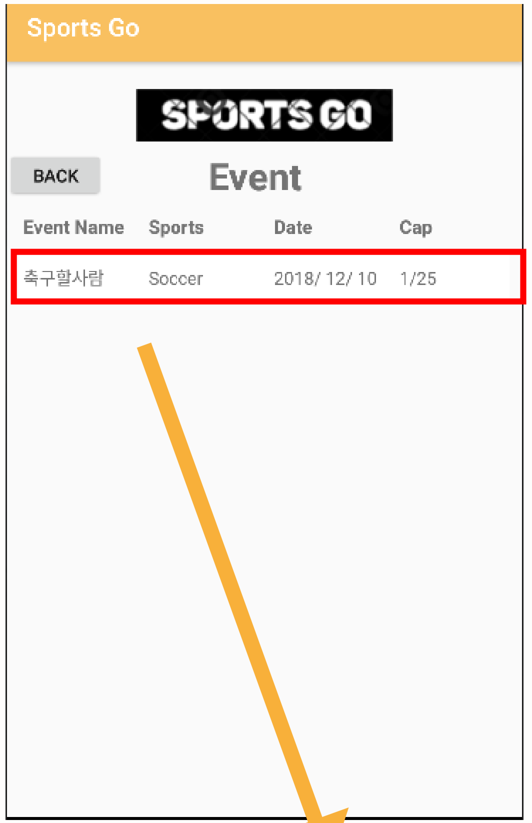


```
public class eventList {
    public String eventName;
    public String eventSports;
    public String eventDate;
    public String eventCapacity;
    public String holder;

    public void setEventName(String e_name) { eventName = e_name; }

    public void setEventSports(String e_Sports) { eventSports = e_Sports; }

    public void setEventDate(String e_Date) { eventDate = e_Date; }

    public String getEventName() {
        return this.eventName;
    }
    public String getEventSports() { return this.eventSports; }

    public String getEventDate() { return this.eventDate; }

    public eventList(){ }

    public eventList(String eventName, String eventSports, String eventDate, String holder, String capacity){
        this.eventName = eventName;
        this.eventSports = eventSports;
        this.eventDate = eventDate;
        this.holder = holder;
        this.eventCapacity = capacity;
    }
}
```

Set each element of the list view with event list information which is saved in the array list(member variable)

## Sports Event Functions

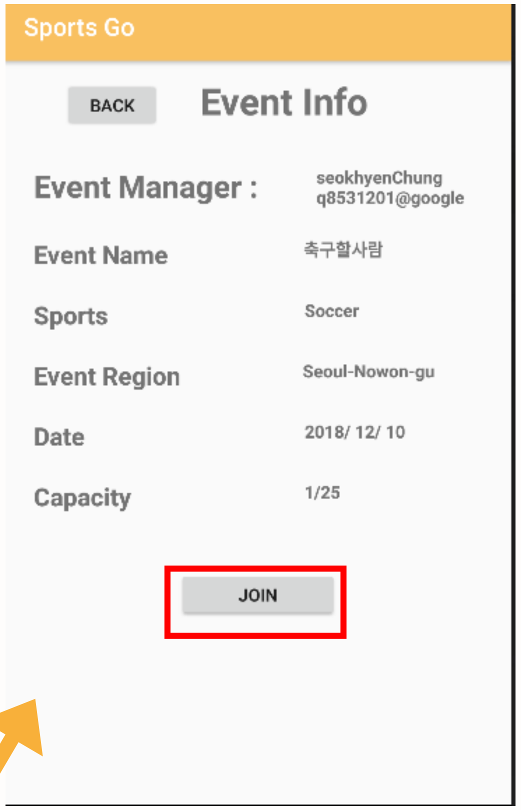
### 5. Join Sports Event


Sports Go

BACK

Event

| Event Name | Sports | Date | Cap |
|---|---|---|---|
| 축구할사람 | Soccer | 2018/ 12/ 10 | 1/25 |

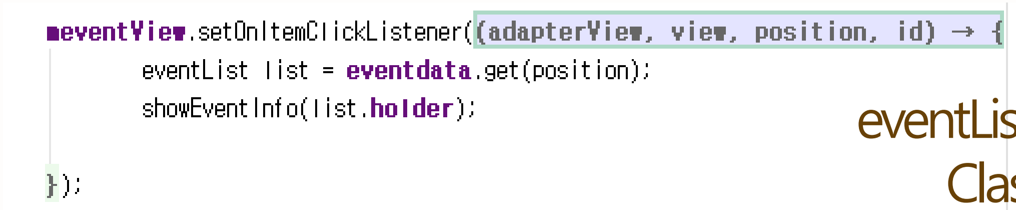

Sports Go

BACK

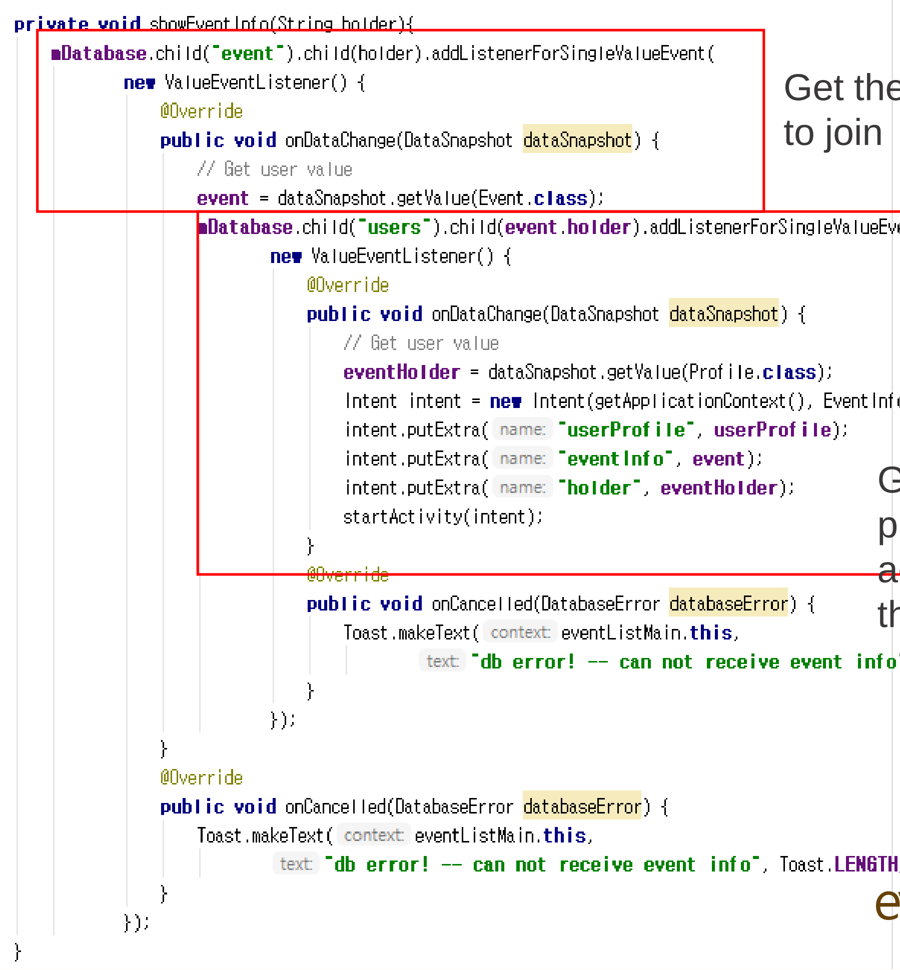
Event Info

| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 1/25 |

JOIN


```
mEventView.setOnItemClickListener((adapterView, view, position, id) → {
        eventList list = eventdata.get(position);
        showEventInfo(list.holder);

});
```

eventListMain Class


```
private void showEventInfo(String holder){
    mDatabase.child("event").child(holder).addListenerForSingleValueEvent(
            new ValueEventListener() {
                @Override
                public void onDataChange(DataSnapshot dataSnapshot) {
                    // Get user value
                    event = dataSnapshot.getValue(Event.class);
                    mDatabase.child("users").child(event.holder).addListenerForSingleValueEvent(
                            new ValueEventListener() {
                                @Override
                                public void onDataChange(DataSnapshot dataSnapshot) {
                                    // Get user value
                                    eventHolder = dataSnapshot.getValue(Profile.class);
                                    Intent intent = new Intent(getApplicationContext(), EventInfo.class);
                                    intent.putExtra( name: "userProfile", userProfile);
                                    intent.putExtra( name: "eventInfo", event);
                                    intent.putExtra( name: "holder", eventHolder);
                                    startActivity(intent);
                                }
                                @Override
                                public void onCancelled(DatabaseError databaseError) {
                                    Toast.makeText( context: eventListMain.this,
                                            text: "db error! -- can not receive event info", Toast.LENGTH_SHORT).show();
                                }
                            });
                }
                @Override
                public void onCancelled(DatabaseError databaseError) {
                    Toast.makeText( context: eventListMain.this,
                            text: "db error! -- can not receive event info", Toast.LENGTH_SHORT).show();
                }
            });
}
```

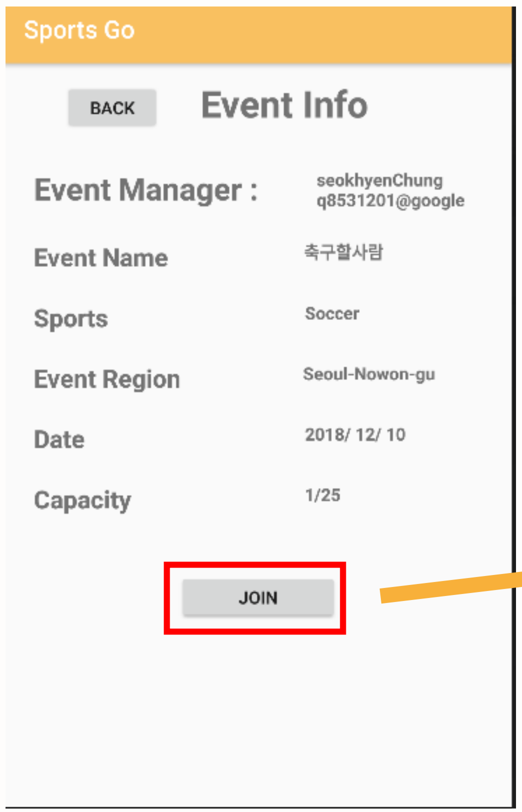
Get the event information to join

Get the event holder profile and start the activity which shows the event details

eventListMain Class

## Sports Event Functions

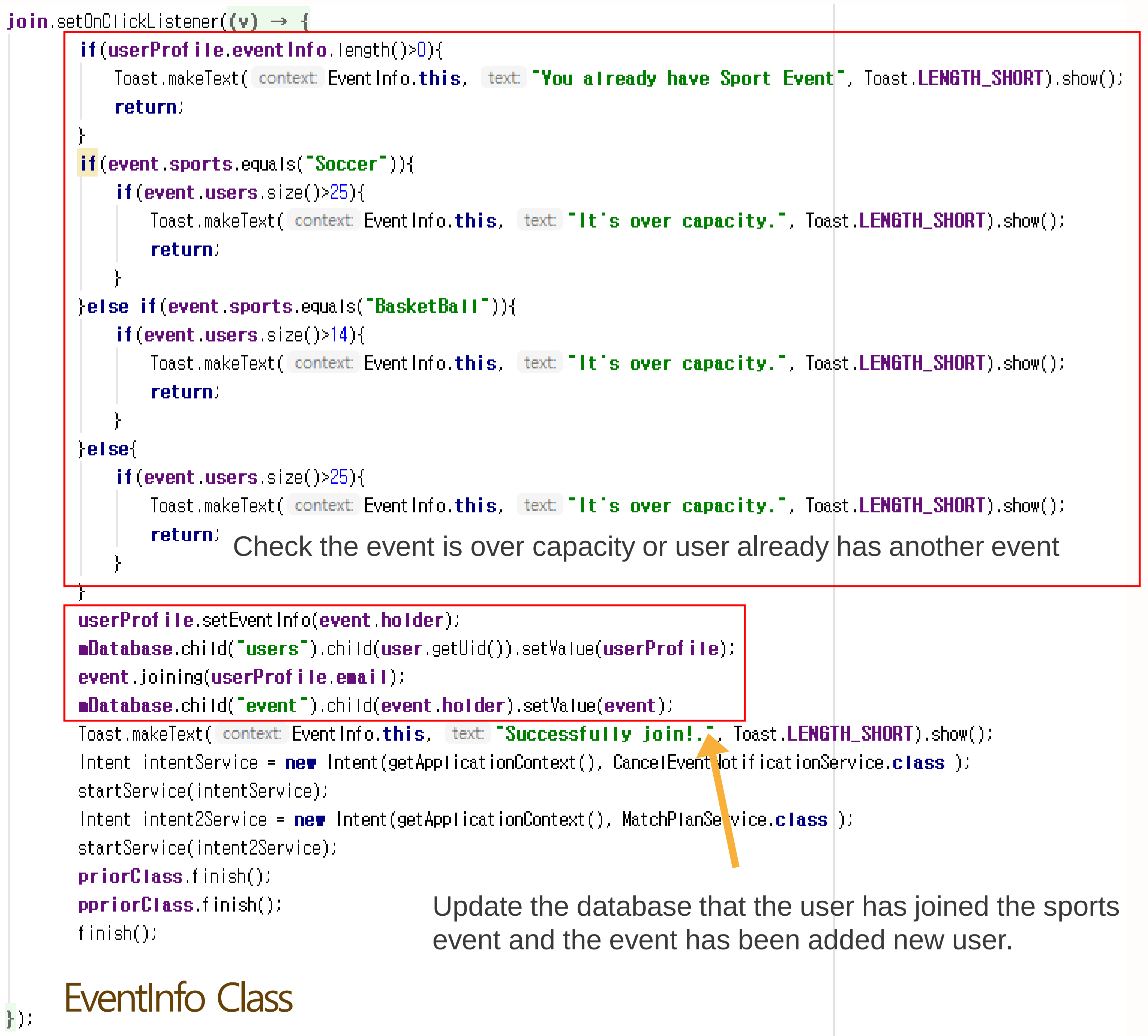
### 5. Join Sports Event


Sports Go

BACK

Event Info

| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 1/25 |


```
join.setOnClickListener((v) → {
        if(userProfile.eventInfo.length()>0){
            Toast.makeText( context: EventInfo.this,  text: "You already have Sport Event", Toast.LENGTH_SHORT).show();
            return;
        }
        if(event.sports.equals("Soccer")){
            if(event.users.size()>25){
                Toast.makeText( context: EventInfo.this,  text: "It's over capacity.", Toast.LENGTH_SHORT).show();
                return;
            }
        }else if(event.sports.equals("BasketBall")){
            if(event.users.size()>14){
                Toast.makeText( context: EventInfo.this,  text: "It's over capacity.", Toast.LENGTH_SHORT).show();
                return;
            }
        }else{
            if(event.users.size()>25){
                Toast.makeText( context: EventInfo.this,  text: "It's over capacity.", Toast.LENGTH_SHORT).show();
                return;
            }
        }
        userProfile.setEventInfo(event.holder);
        mDatabase.child("users").child(user.getUid()).setValue(userProfile);
        event.joining(userProfile.email);
        mDatabase.child("event").child(event.holder).setValue(event);
        Toast.makeText( context: EventInfo.this,  text: "Successfully join!.", Toast.LENGTH_SHORT).show();
        Intent intentService = new Intent(getApplicationContext(), CancelEventNotificationService.class );
        startService(intentService);
        Intent intent2Service = new Intent(getApplicationContext(), MatchPlanService.class );
        startService(intent2Service);
        priorClass.finish();
        ppriorClass.finish();
        finish();
});
```

Check the event is over capacity or user already has another event

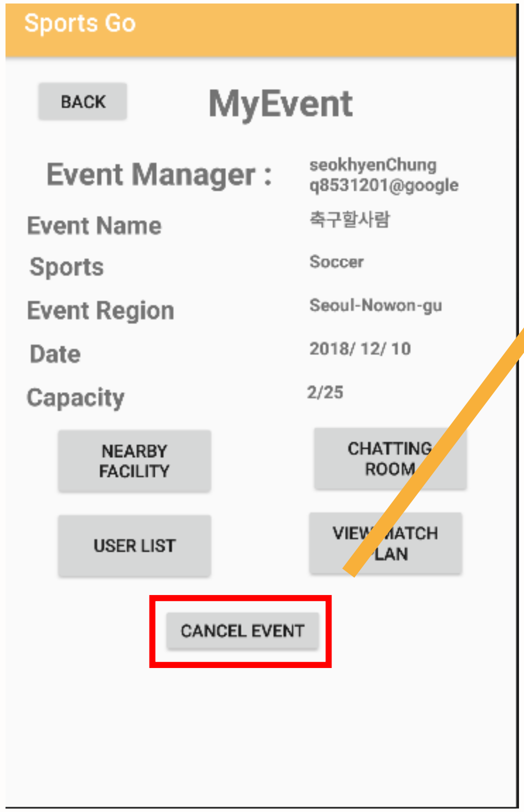
Update the database that the user has joined the sports event and the event has been added new user.

EventInfo Class

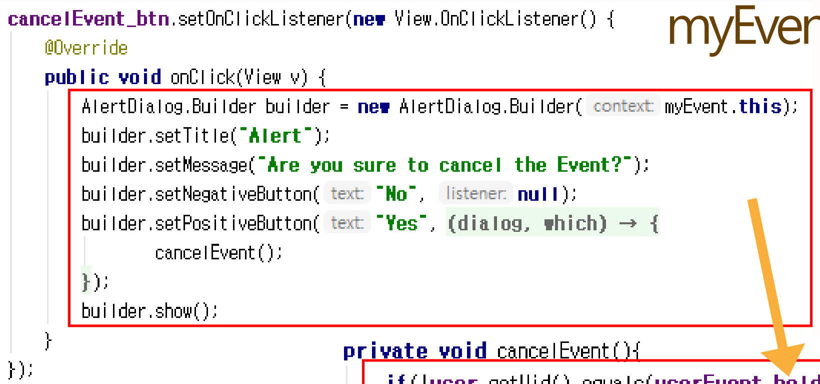
03 Main Functions – Source code

## Sports Event Functions

### 6. Cancel Sports Event



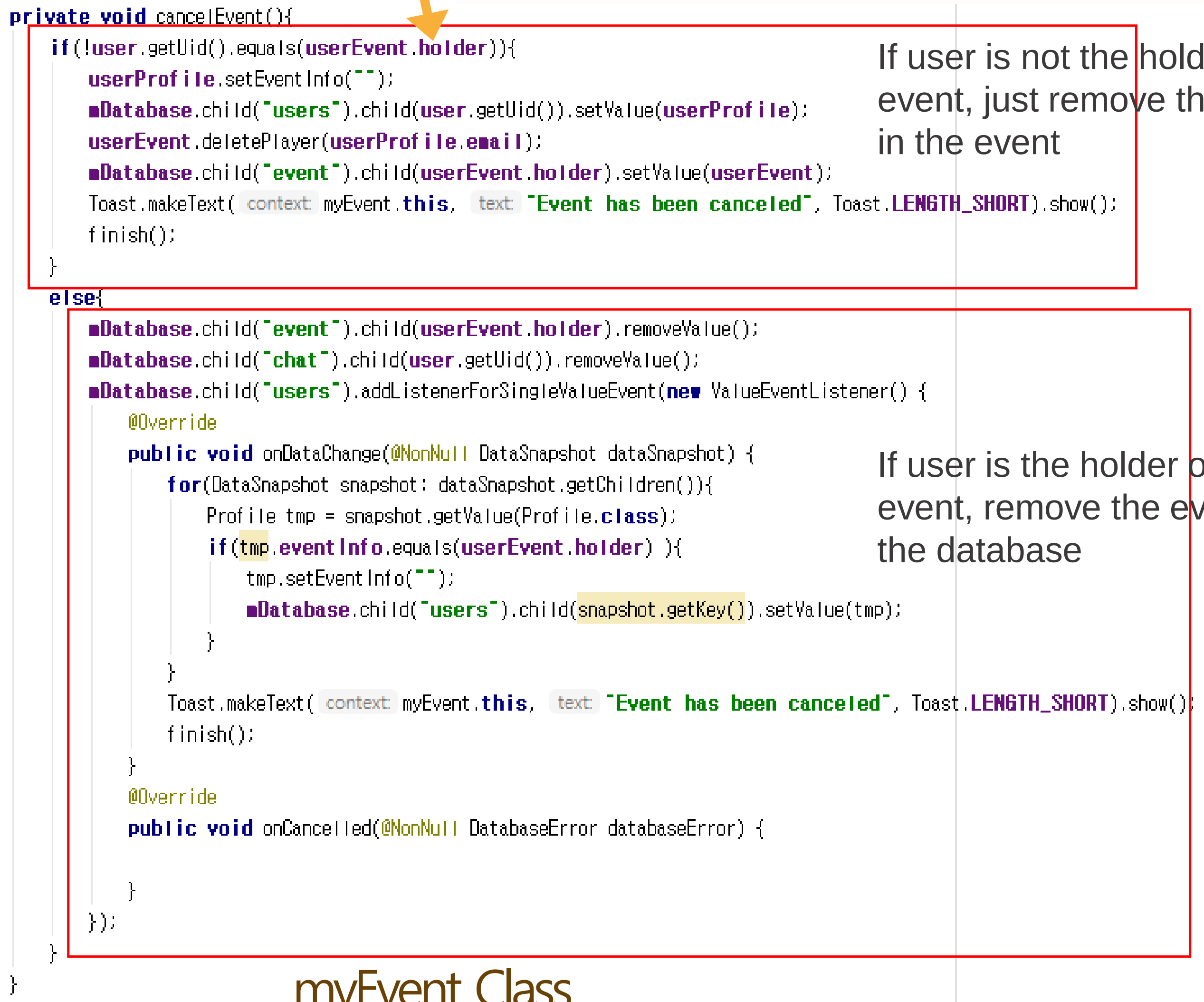
Event Manager : seokhyenChung q8531201@google
Event Name 축구할사람
Sports Soccer
Event Region Seoul-Nowon-gu
Date 2018/ 12/ 10
Capacity 2/25



```
cancelEvent_btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        AlertDialog.Builder builder = new AlertDialog.Builder( context: myEvent.this);
        builder.setTitle("Alert");
        builder.setMessage("Are you sure to cancel the Event?");
        builder.setNegativeButton( text: "No", listener: null);
        builder.setPositiveButton( text: "Yes", (dialog, which) → {
                cancelEvent();
        });
        builder.show();
    }
});
```

myEvent Class

Show the confirmation dialog



```
private void cancelEvent(){
    if(!user.getUid().equals(userEvent.holder)){
        userProfile.setEventInfo("");
        mDatabase.child("users").child(user.getUid()).setValue(userProfile);
        userEvent.deletePlayer(userProfile.email);
        mDatabase.child("event").child(userEvent.holder).setValue(userEvent);
        Toast.makeText( context: myEvent.this, text: "Event has been canceled", Toast.LENGTH_SHORT).show();
        finish();
    }
    else{
        mDatabase.child("event").child(userEvent.holder).removeValue();
        mDatabase.child("chat").child(user.getUid()).removeValue();
        mDatabase.child("users").addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for(DataSnapshot snapshot: dataSnapshot.getChildren()){
                    Profile tmp = snapshot.getValue(Profile.class);
                    if(tmp.eventInfo.equals(userEvent.holder) ){
                        tmp.setEventInfo("");
                        mDatabase.child("users").child(snapshot.getKey()).setValue(tmp);
                    }
                }
                Toast.makeText( context: myEvent.this, text: "Event has been canceled", Toast.LENGTH_SHORT).show();
                finish();
            }
            @Override
            public void onCancelled(@NonNull DatabaseError databaseError) {

            }
        });
    }
}
```

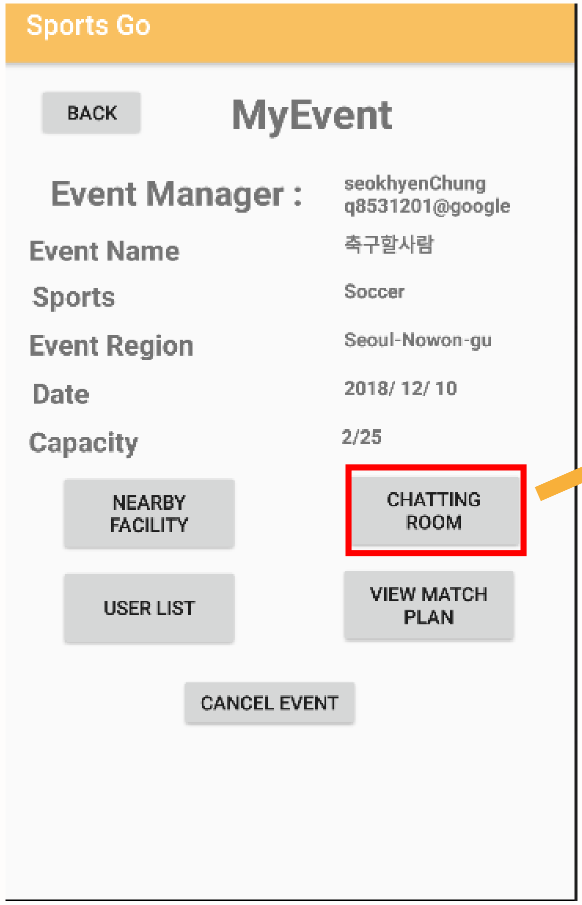
If user is not the holder of the event, just remove the player in the event

If user is the holder of the event, remove the event in the database

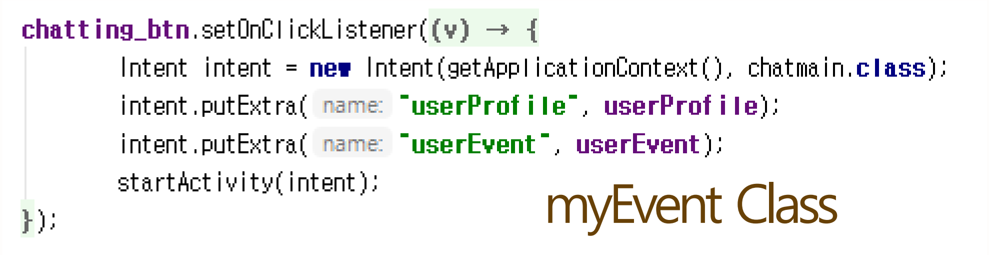
myEvent Class

## Sports Event Functions

### 7. Chatting – Using Firebase Realtime Database



| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 2/25 |



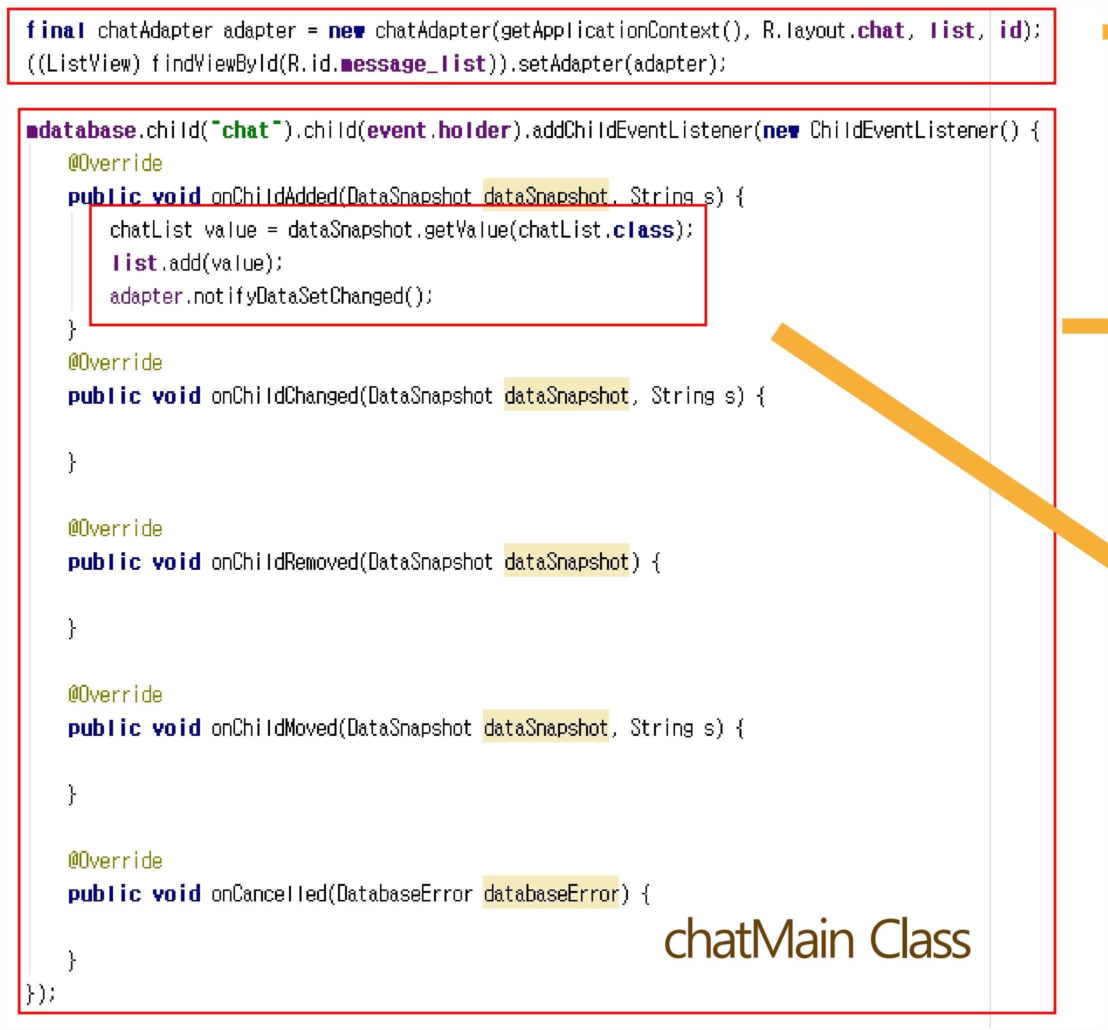
```
chatting_btn.setOnClickListener((v) → {
        Intent intent = new Intent(getApplicationContext(), chatmain.class);
        intent.putExtra( name: "userProfile", userProfile);
        intent.putExtra( name: "userEvent", userEvent);
        startActivity(intent);
});
```

myEvent Class

With the user profile and the event information user taken, start the chatting activity.

## Sports Event Functions

### 7. Chatting – Using Firebase Realtime Database



```java
final chatAdapter adapter = new chatAdapter(getApplicationContext(), R.layout.chat, list, id);
((ListView) findViewById(R.id.message_list)).setAdapter(adapter);

mdatabase.child("chat").child(event.holder).addChildEventListener(new ChildEventListener() {
    @Override
    public void onChildAdded(DataSnapshot dataSnapshot, String s) {
        chatList value = dataSnapshot.getValue(chatList.class);
        list.add(value);
        adapter.notifyDataSetChanged();
    }
    @Override
    public void onChildChanged(DataSnapshot dataSnapshot, String s) {

    }

    @Override
    public void onChildRemoved(DataSnapshot dataSnapshot) {

    }

    @Override
    public void onChildMoved(DataSnapshot dataSnapshot, String s) {

    }

    @Override
    public void onCancelled(DatabaseError databaseError) {

    }
});
```

chatMain Class

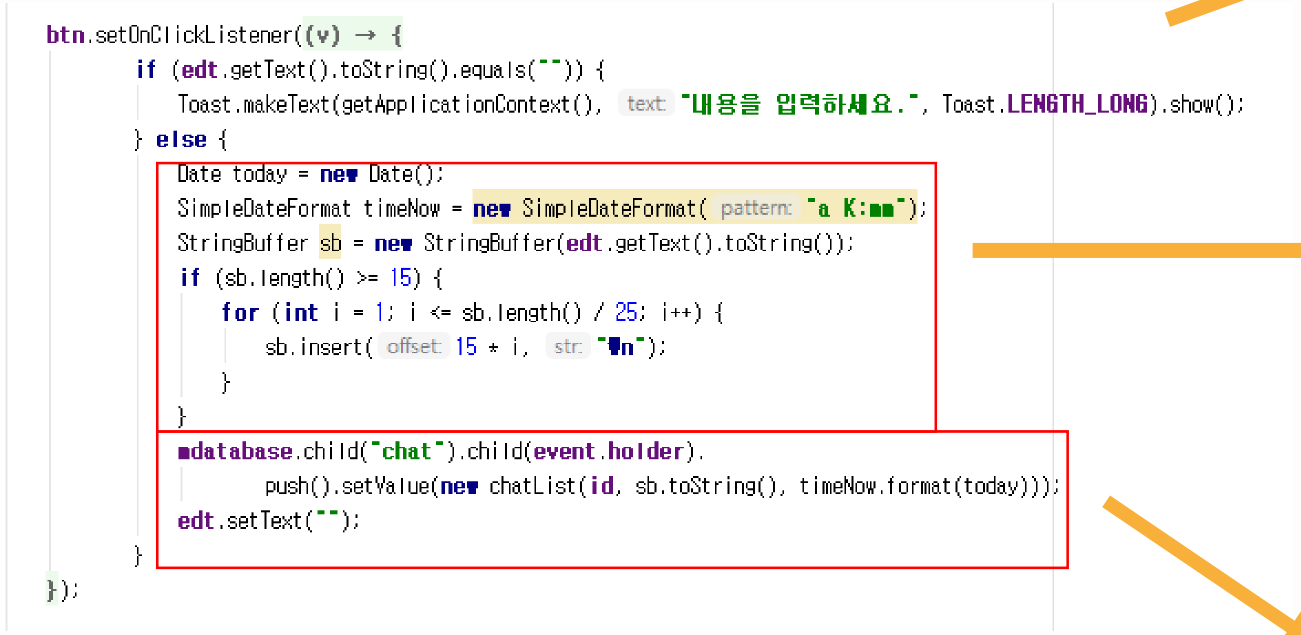
Adapter and Listview object for chatting view

Database change listener for chatting data for user sports event

If this listener detects the chat data has been added, add that chat data to the list and notify to list view that some chat data has been added. Then, list view show that.

## Sports Event Functions

### 7. Chatting – Using Firebase Realtime Database



```
btn.setOnClickListener((v) → {
        if (edt.getText().toString().equals("")) {
            Toast.makeText(getApplicationContext(), text: "내용을 입력하세요.", Toast.LENGTH_LONG).show();
        } else {
            Date today = new Date();
            SimpleDateFormat timeNow = new SimpleDateFormat( pattern: "a K:mm");
            StringBuffer sb = new StringBuffer(edt.getText().toString());
            if (sb.length() >= 15) {
                for (int i = 1; i <= sb.length() / 25; i++) {
                    sb.insert( offset: 15 * i, str: "₩n");
                }
            }
            mdatabase.child("chat").child(event.holder).
                    push().setValue(new chatList(id, sb.toString(), timeNow.format(today)));
            edt.setText("");
        }
});
```

Method that send the chat message from user input

Set current time into date format and new line in the chat message

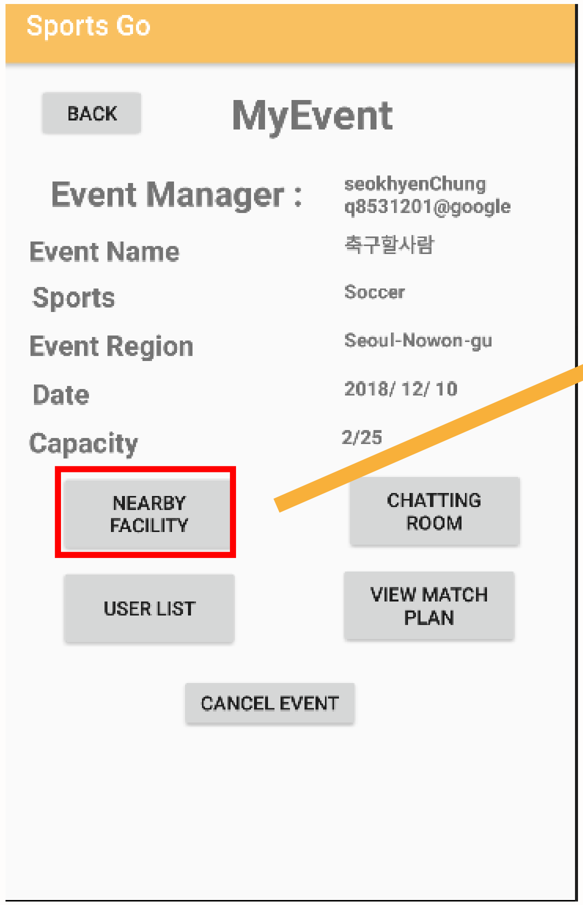
Add new chat message from user to the firebase database

chatMain Class

03 Main Functions – Source code

## Sports Event Functions

### 8. Nearby Facility Information

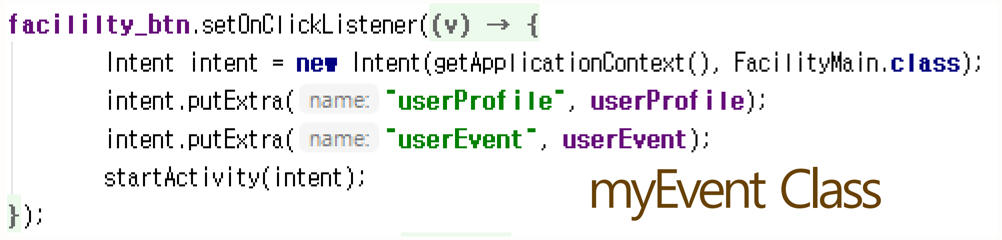

Sports Go

BACK

MyEvent

Event Manager : seokhyenChung q8531201@google

| | |
|---|---|
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 2/25 |

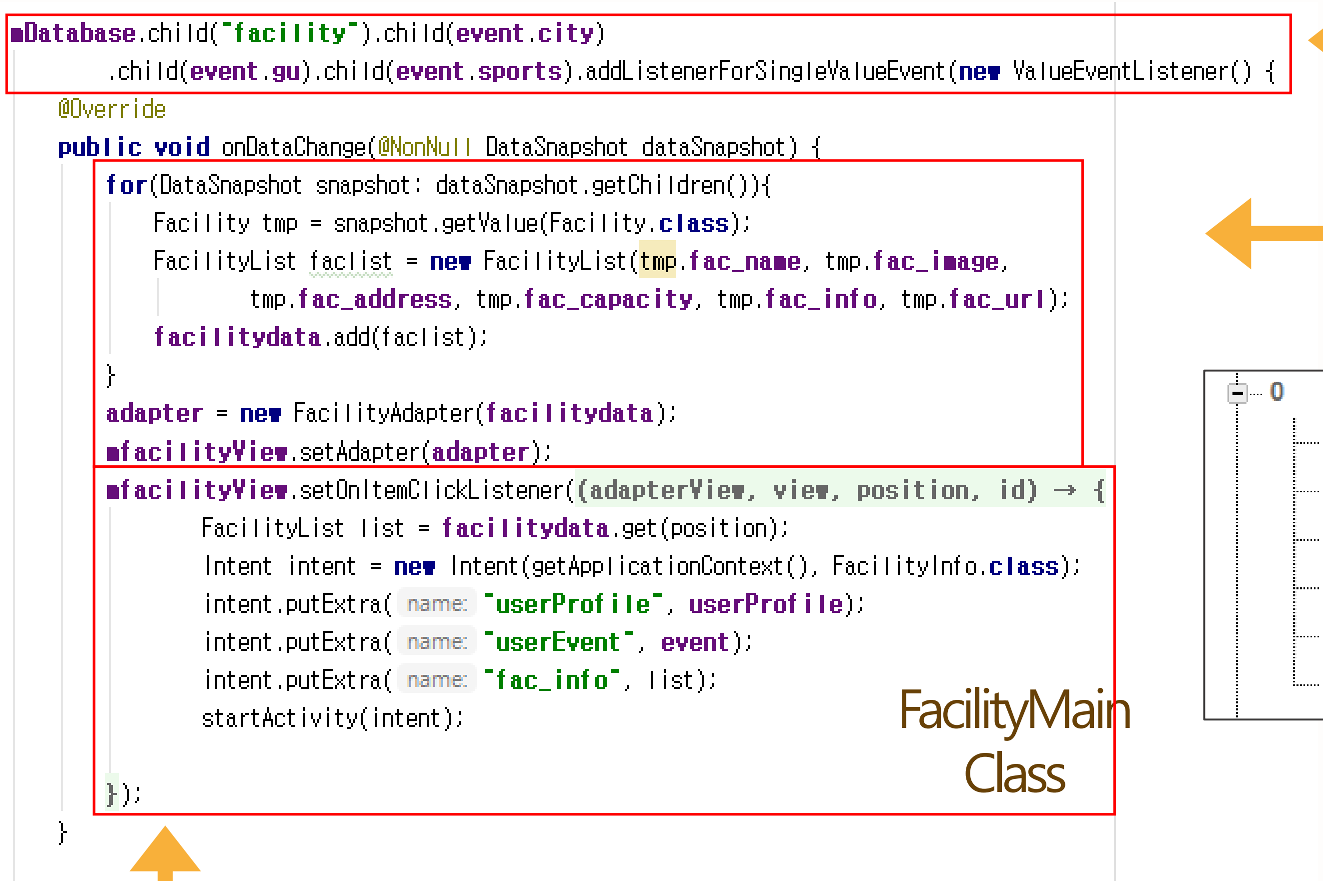

```
facility_btn.setOnClickListener((v) -> {
        Intent intent = new Intent(getApplicationContext(), FacilityMain.class);
        intent.putExtra( name: "userProfile", userProfile);
        intent.putExtra( name: "userEvent", userEvent);
        startActivity(intent);
});
```

myEvent Class

With the user profile and the event information user taken, start the nearby facility information activity.

## Sports Event Functions

### 8. Nearby Facility Information



```java
mDatabase.child("facility").child(event.city)
        .child(event.gu).child(event.sports).addListenerForSingleValueEvent(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        for(DataSnapshot snapshot: dataSnapshot.getChildren()){
            Facility tmp = snapshot.getValue(Facility.class);
            FacilityList faclist = new FacilityList(tmp.fac_name, tmp.fac_image,
                    tmp.fac_address, tmp.fac_capacity, tmp.fac_info, tmp.fac_url);
            facilitydata.add(faclist);
        }
        adapter = new FacilityAdapter(facilitydata);
        mfacilityView.setAdapter(adapter);
        mfacilityView.setOnItemClickListener((adapterView, view, position, id) -> {
                FacilityList list = facilitydata.get(position);
                Intent intent = new Intent(getApplicationContext(), FacilityInfo.class);
                intent.putExtra( name: "userProfile", userProfile);
                intent.putExtra( name: "userEvent", event);
                intent.putExtra( name: "fac_info", list);
                startActivity(intent);

        });
    }
```

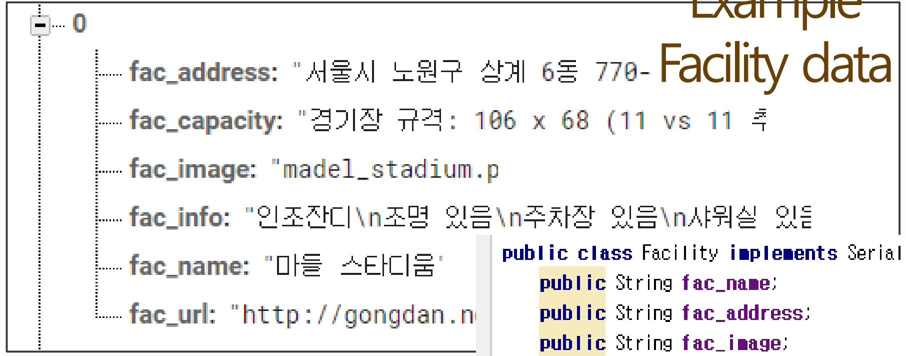
FacilityMain Class

Read all facilities according to the event region from database

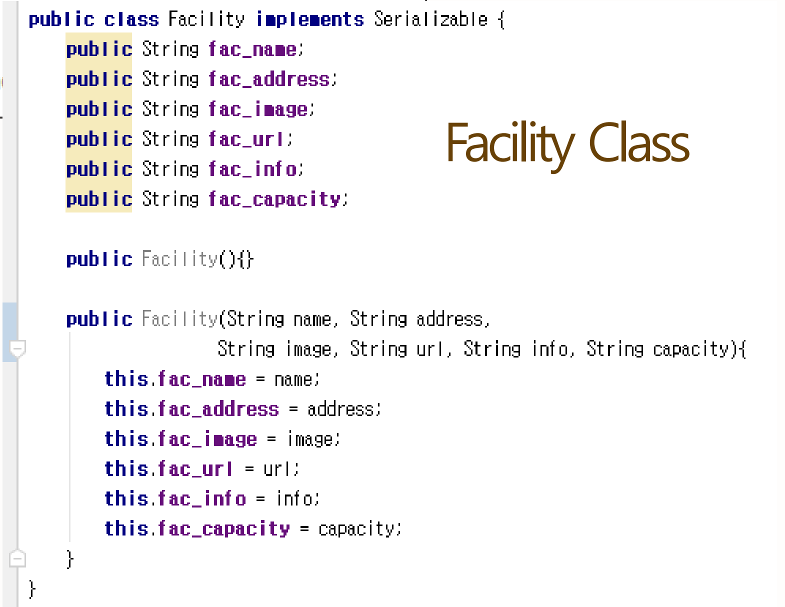
Add facilities information to array list. Then, show the facility list in list view.

Listview onclick Listener that shows the detail information of facility.

Example Facility data



```
0
    fac_address: "서울시 노원구 상계 6동 770-
    fac_capacity: "경기장 규격: 106 x 68 (11 vs 11 축
    fac_image: "madel_stadium.p
    fac_info: "인조잔디\n조명 있음\n주차장 있음\n샤워실 있
    fac_name: "마들 스타디움"
    fac_url: "http://gongdan.n
```



```java
public class Facility implements Serializable {
    public String fac_name;
    public String fac_address;
    public String fac_image;
    public String fac_url;
    public String fac_info;
    public String fac_capacity;

    public Facility(){}

    public Facility(String name, String address,
                    String image, String url, String info, String capacity){
        this.fac_name = name;
        this.fac_address = address;
        this.fac_image = image;
        this.fac_url = url;
        this.fac_info = info;
        this.fac_capacity = capacity;
    }
}
```

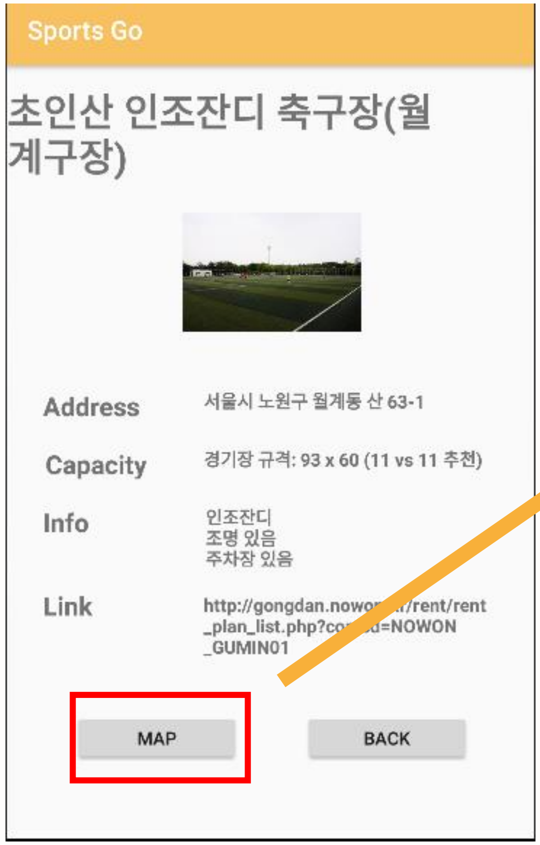
Facility Class

03 Main Functions – Source code

## Sports Event Functions

### 8. Nearby Facility Information – Map Function

***We use Google map API***

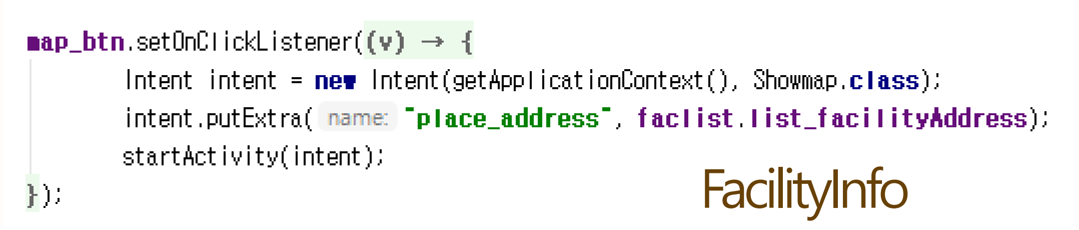


Sports Go

초인산 인조잔디 축구장(월계구장)

| | |
|---|---|
| Address | 서울시 노원구 월계동 산 63-1 |
| Capacity | 경기장 규격: 93 x 60 (11 vs 11 추천) |
| Info | 인조잔디<br>조명 있음<br>주차장 있음 |
| Link | http://gongdan.nowo[illegible]/rent/rent_plan_list.php?co[illegible]d=NOWON_GUMIN01 |

MAP BACK



```
map_btn.setOnClickListener((v) → {
        Intent intent = new Intent(getApplicationContext(), Showmap.class);
        intent.putExtra( name: "place_address", faclist.list_facilityAddress);
        startActivity(intent);
});
```

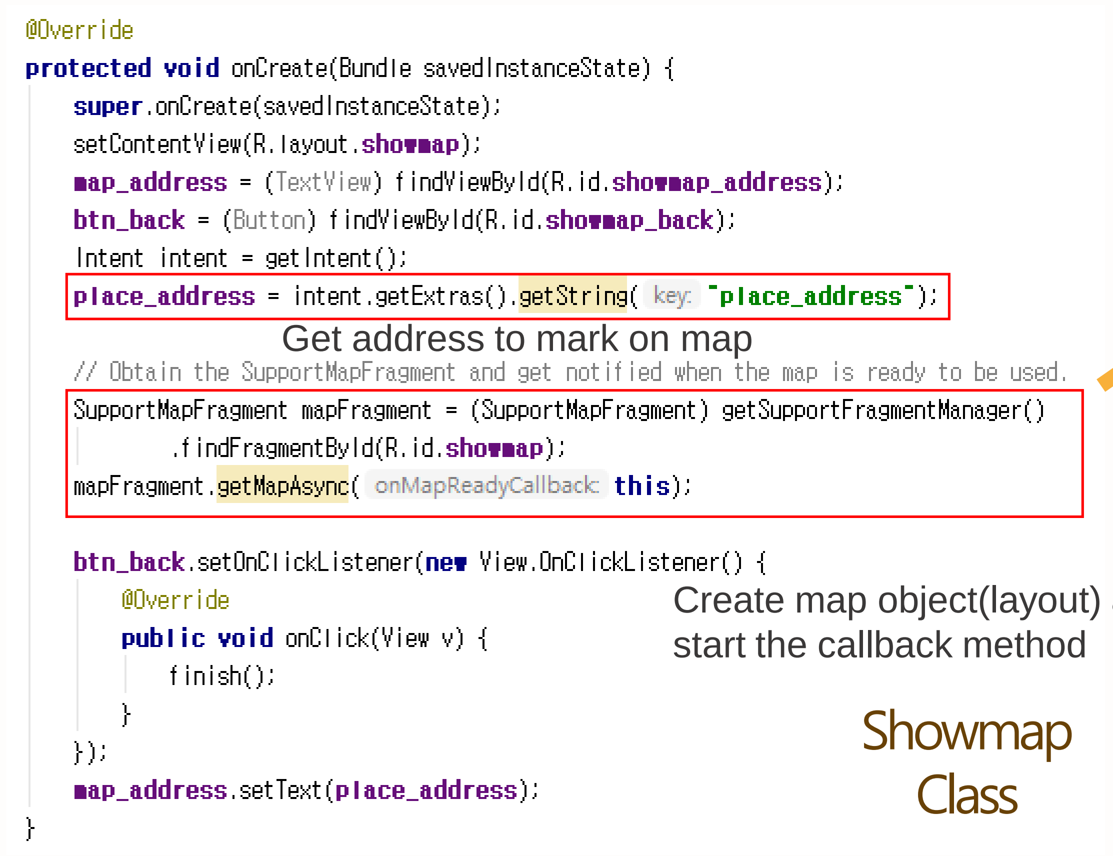
FacilityInfo Class

With the address of facility, start the Showmap activity.

03 Main Functions – Source code

# Sports Event Functions

## 8. Nearby Facility Information – Map Function

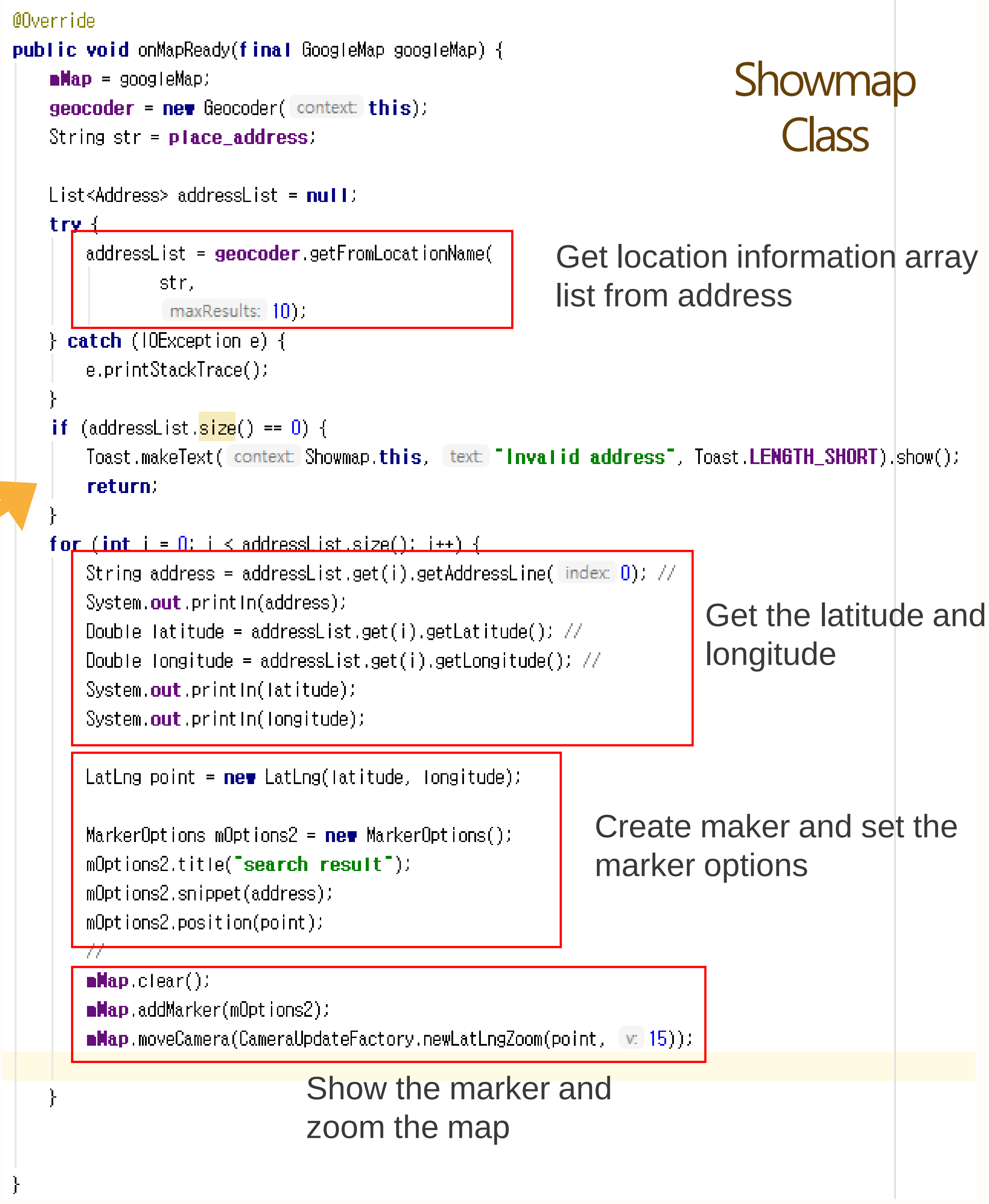


```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.showmap);
    map_address = (TextView) findViewById(R.id.showmap_address);
    btn_back = (Button) findViewById(R.id.showmap_back);
    Intent intent = getIntent();
    place_address = intent.getExtras().getString( key: "place_address");
    // Obtain the SupportMapFragment and get notified when the map is ready to be used.
    SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.showmap);
    mapFragment.getMapAsync( onMapReadyCallback: this);

    btn_back.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();
        }
    });
    map_address.setText(place_address);
}
```

Get address to mark on map

Create map object(layout) and start the callback method

Showmap Class



```
@Override
public void onMapReady(final GoogleMap googleMap) {
    mMap = googleMap;
    geocoder = new Geocoder( context: this);
    String str = place_address;

    List<Address> addressList = null;
    try {
        addressList = geocoder.getFromLocationName(
                str,
                 maxResults: 10);
    } catch (IOException e) {
        e.printStackTrace();
    }
    if (addressList.size() == 0) {
        Toast.makeText( context: Showmap.this,  text: "Invalid address", Toast.LENGTH_SHORT).show();
        return;
    }
    for (int i = 0; i < addressList.size(); i++) {
        String address = addressList.get(i).getAddressLine( index: 0); //
        System.out.println(address);
        Double latitude = addressList.get(i).getLatitude(); //
        Double longitude = addressList.get(i).getLongitude(); //
        System.out.println(latitude);
        System.out.println(longitude);

        LatLng point = new LatLng(latitude, longitude);

        MarkerOptions mOptions2 = new MarkerOptions();
        mOptions2.title("search result");
        mOptions2.snippet(address);
        mOptions2.position(point);
        //
        mMap.clear();
        mMap.addMarker(mOptions2);
        mMap.moveCamera(CameraUpdateFactory.newLatLngZoom(point,  v: 15));

    }

}
```

Showmap Class

Get location information array list from address

Get the latitude and longitude

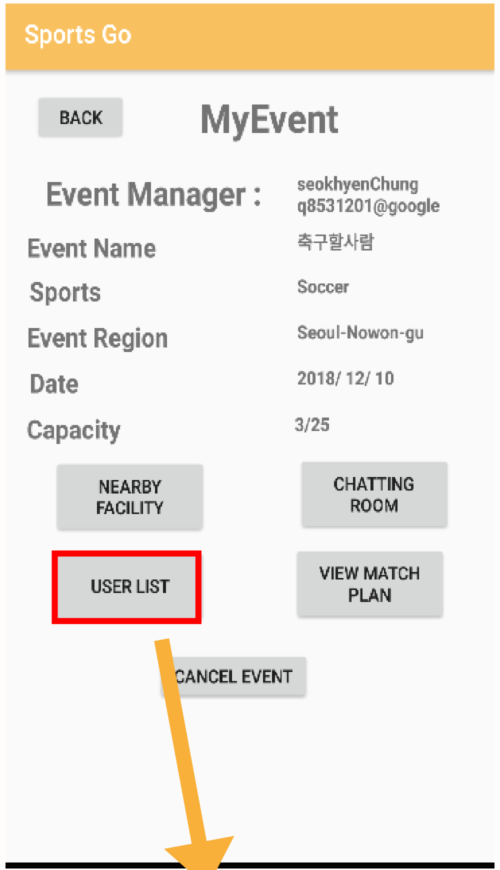
Create maker and set the marker options

Show the marker and zoom the map

03 Main Functions – Source code

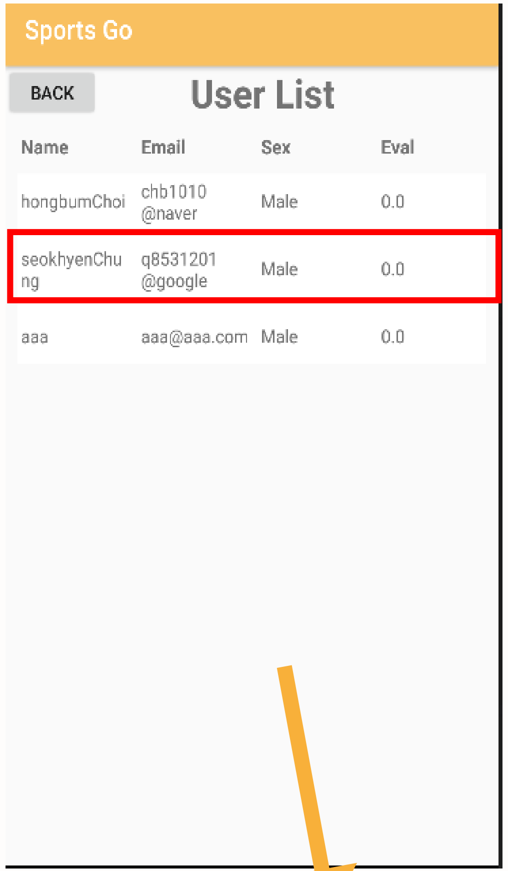
## Sports Event Functions

### 9. User Evaluation


Sports Go

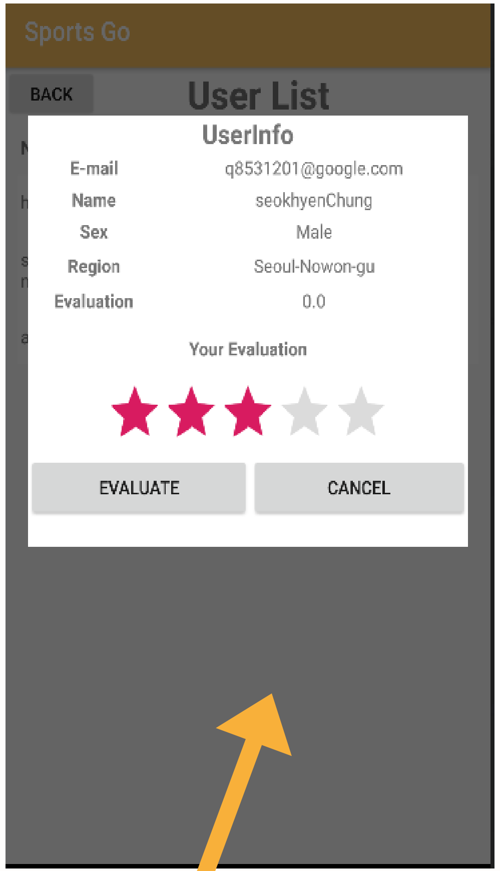
BACK MyEvent

| | |
|---|---|
| Event Manager : | seokhyenChung q8531201@google |
| Event Name | 축구할사람 |
| Sports | Soccer |
| Event Region | Seoul-Nowon-gu |
| Date | 2018/ 12/ 10 |
| Capacity | 3/25 |


Sports Go

BACK User List

| Name | Email | Sex | Eval |
|---|---|---|---|
| hongbumChoi | chb1010 @naver | Male | 0.0 |
| seokhyenChung | q8531201 @google | Male | 0.0 |
| aaa | aaa@aaa.com | Male | 0.0 |


Sports Go

BACK User List

UserInfo

| | |
|---|---|
| E-mail | q8531201@google.com |
| Name | seokhyenChung |
| Sex | Male |
| Region | Seoul-Nowon-gu |
| Evaluation | 0.0 |

Your Evaluation

```
userList_btn.setOnClickListener((v) → {
        Intent intent = new Intent(getApplicationContext(), UserListMain.class);
        intent.putExtra( name: "userProfile", userProfile);
        intent.putExtra( name: "userEvent", userEvent);
        startActivity(intent);
});
```

myEvent Class

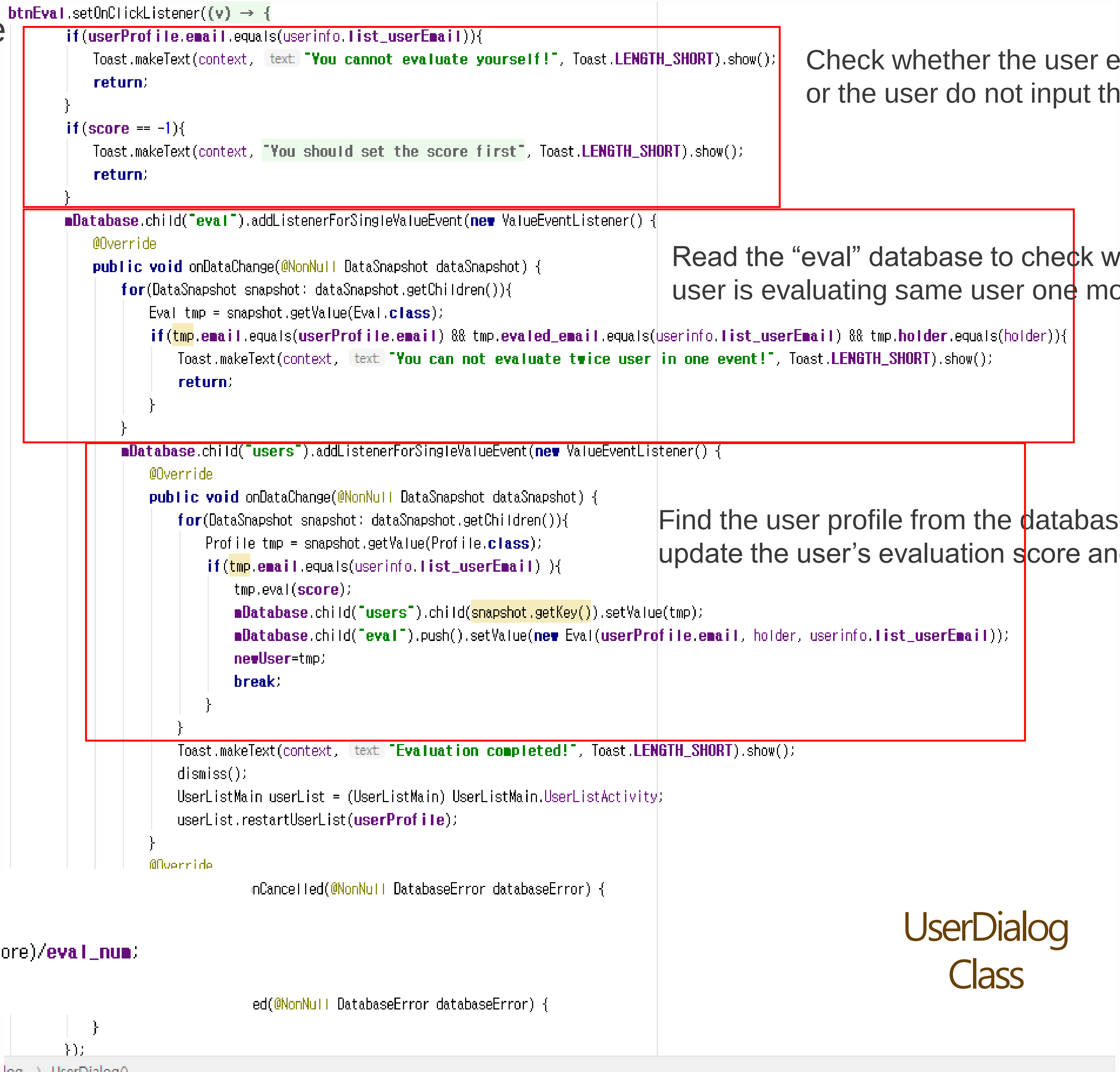
```
private void showUserInfo(UserList userinfo){
    DisplayMetrics dm = getApplicationContext().getResources().getDisplayMetrics();
    int width = dm.widthPixels;
    int height = dm.heightPixels;
    ud = new UserDialog( context: UserListMain.this, userinfo, userProfile, event.holder);
    WindowManager.LayoutParams wm = ud.getWindow().getAttributes();
    wm.copyFrom(ud.getWindow().getAttributes());
    wm.width = width-100;
    wm.height = height;
    ud.show();

}
```

UserListMain Class

03 Main Functions – Source code

## Sports Event Functions

### 9. User Evaluation



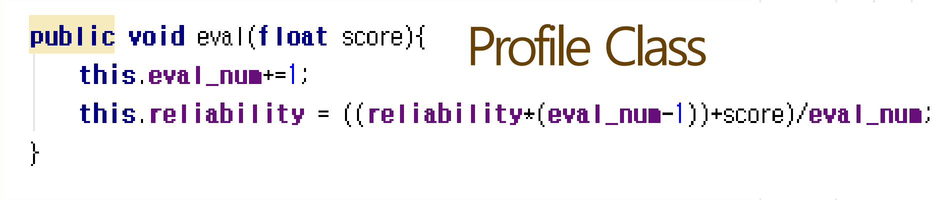
```
btnEval.setOnClickListener((v) -> {
        if(userProfile.email.equals(userinfo.list_userEmail)){
            Toast.makeText(context, text: "You cannot evaluate yourself!", Toast.LENGTH_SHORT).show();
            return;
        }
        if(score == -1){
            Toast.makeText(context, "You should set the score first", Toast.LENGTH_SHORT).show();
            return;
        }
        mDatabase.child("eval").addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for(DataSnapshot snapshot: dataSnapshot.getChildren()){
                    Eval tmp = snapshot.getValue(Eval.class);
                    if(tmp.email.equals(userProfile.email) && tmp.evaled_email.equals(userinfo.list_userEmail) && tmp.holder.equals(holder)){
                        Toast.makeText(context, text: "You can not evaluate twice user in one event!", Toast.LENGTH_SHORT).show();
                        return;
                    }
                }
                mDatabase.child("users").addListenerForSingleValueEvent(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                        for(DataSnapshot snapshot: dataSnapshot.getChildren()){
                            Profile tmp = snapshot.getValue(Profile.class);
                            if(tmp.email.equals(userinfo.list_userEmail) ){
                                tmp.eval(score);
                                mDatabase.child("users").child(snapshot.getKey()).setValue(tmp);
                                mDatabase.child("eval").push().setValue(new Eval(userProfile.email, holder, userinfo.list_userEmail));
                                newUser=tmp;
                                break;
                            }
                        }
                        Toast.makeText(context, text: "Evaluation completed!", Toast.LENGTH_SHORT).show();
                        dismiss();
                        UserListMain userList = (UserListMain) UserListMain.UserListActivity;
                        userList.restartUserList(userProfile);
                    }
                    @Override
                    nCancelled(@NonNull DatabaseError databaseError) {

                ed(@NonNull DatabaseError databaseError) {
            }
        });
```

Check whether the user evaluate himself or the user do not input the score

Read the "eval" database to check whether the user is evaluating same user one more

Find the user profile from the database and update the user's evaluation score and save it

UserDialog Class



```
public void eval(float score){
    this.eval_num+=1;
    this.reliability = ((reliability*(eval_num-1))+score)/eval_num;
}
```

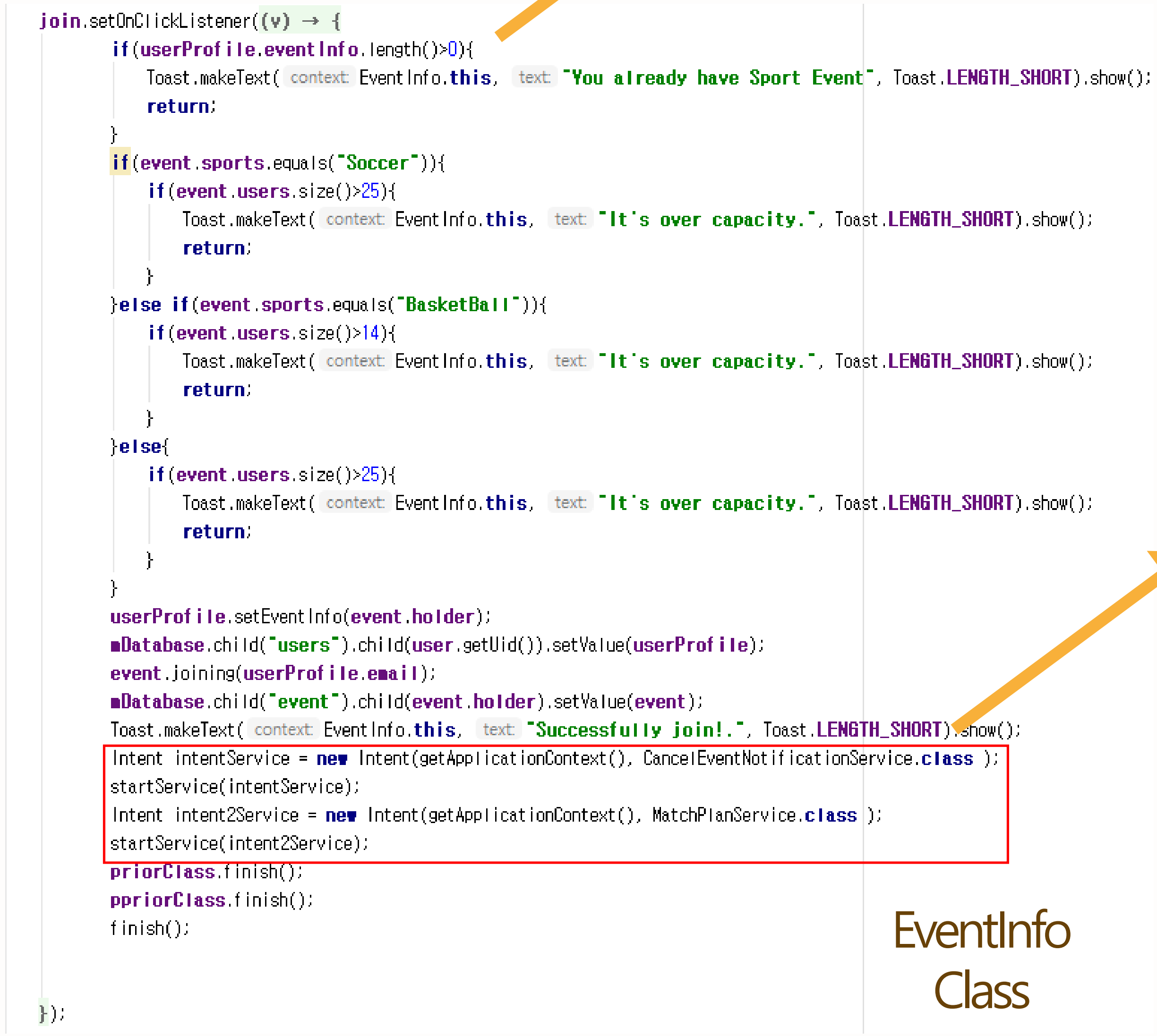
Profile Class

03 Main Functions – Source code

## Alarm service

### 10. Alarm Services

***Event Join method!***



```
join.setOnClickListener((v) → {
        if(userProfile.eventInfo.length()>0){
            Toast.makeText( context: EventInfo.this,  text: "You already have Sport Event", Toast.LENGTH_SHORT).show();
            return;
        }
        if(event.sports.equals("Soccer")){
            if(event.users.size()>25){
                Toast.makeText( context: EventInfo.this,  text: "It's over capacity.", Toast.LENGTH_SHORT).show();
                return;
            }
        }else if(event.sports.equals("BasketBall")){
            if(event.users.size()>14){
                Toast.makeText( context: EventInfo.this,  text: "It's over capacity.", Toast.LENGTH_SHORT).show();
                return;
            }
        }else{
            if(event.users.size()>25){
                Toast.makeText( context: EventInfo.this,  text: "It's over capacity.", Toast.LENGTH_SHORT).show();
                return;
            }
        }
        userProfile.setEventInfo(event.holder);
        mDatabase.child("users").child(user.getUid()).setValue(userProfile);
        event.joining(userProfile.email);
        mDatabase.child("event").child(event.holder).setValue(event);
        Toast.makeText( context: EventInfo.this,  text: "Successfully join!.", Toast.LENGTH_SHORT).show();
        Intent intentService = new Intent(getApplicationContext(), CancelEventNotificationService.class );
        startService(intentService);
        Intent intent2Service = new Intent(getApplicationContext(), MatchPlanService.class );
        startService(intent2Service);
        priorClass.finish();
        ppriorClass.finish();
        finish();


});
```

EventInfo Class

When user has joined the sports event, the two service would start to run.

One is Cancel Alarm service which notify the user when the event has been canceled.

The other is Match Plan Service which notify the user when the match plan has been registered in sports event.

03 Main Functions – Source code

## Alarm service

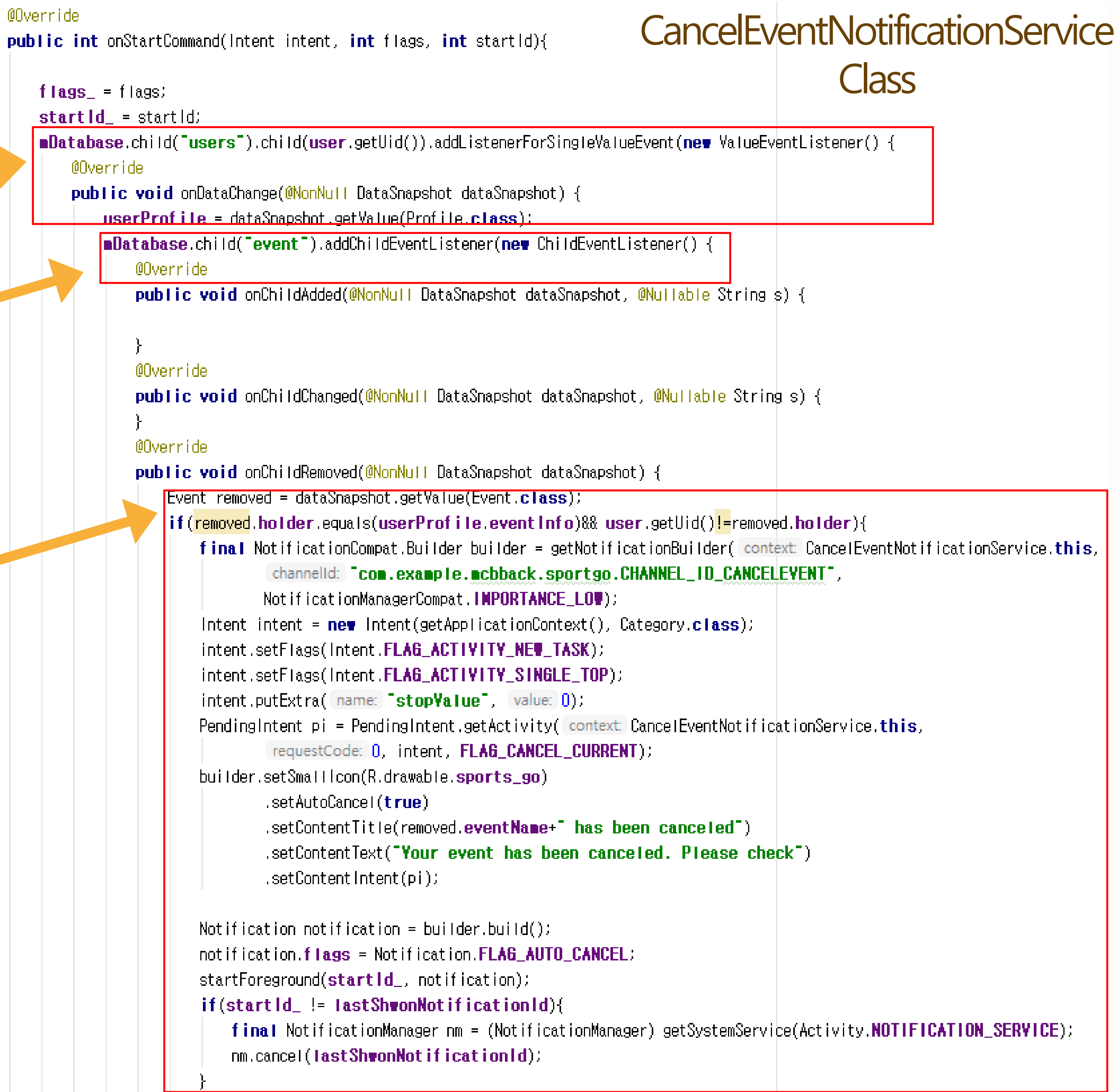
### 10. Alarm Services - Event Cancel Alarm

## CancelEventNotificationService Class

Get the user profile

Add event database listener that detect remove of events

When the listener detects the remove of events, check that the removed event is the event that user taken. If true, the service send notification to the user device.



```
@Override
public int onStartCommand(Intent intent, int flags, int startId){

    flags_ = flags;
    startId_ = startId;
    mDatabase.child("users").child(user.getUid()).addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            userProfile = dataSnapshot.getValue(Profile.class);
            mDatabase.child("event").addChildEventListener(new ChildEventListener() {
                @Override
                public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

                }
                @Override
                public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {
                }
                @Override
                public void onChildRemoved(@NonNull DataSnapshot dataSnapshot) {
                    Event removed = dataSnapshot.getValue(Event.class);
                    if(removed.holder.equals(userProfile.eventInfo)&& user.getUid()!=removed.holder){
                        final NotificationCompat.Builder builder = getNotificationBuilder( context: CancelEventNotificationService.this,
                                channelId: "com.example.mcbback.sportgo.CHANNEL_ID_CANCELEVENT",
                                NotificationManagerCompat.IMPORTANCE_LOW);
                        Intent intent = new Intent(getApplicationContext(), Category.class);
                        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                        intent.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
                        intent.putExtra( name: "stopValue", value: 0);
                        PendingIntent pi = PendingIntent.getActivity( context: CancelEventNotificationService.this,
                                requestCode: 0, intent, FLAG_CANCEL_CURRENT);
                        builder.setSmallIcon(R.drawable.sports_go)
                                .setAutoCancel(true)
                                .setContentTitle(removed.eventName+" has been canceled")
                                .setContentText("Your event has been canceled. Please check")
                                .setContentIntent(pi);

                        Notification notification = builder.build();
                        notification.flags = Notification.FLAG_AUTO_CANCEL;
                        startForeground(startId_, notification);
                        if(startId_ != lastShwonNotificationId){
                            final NotificationManager nm = (NotificationManager) getSystemService(Activity.NOTIFICATION_SERVICE);
                            nm.cancel(lastShwonNotificationId);
                        }
```

03 Main Functions – Source code

## Alarm service

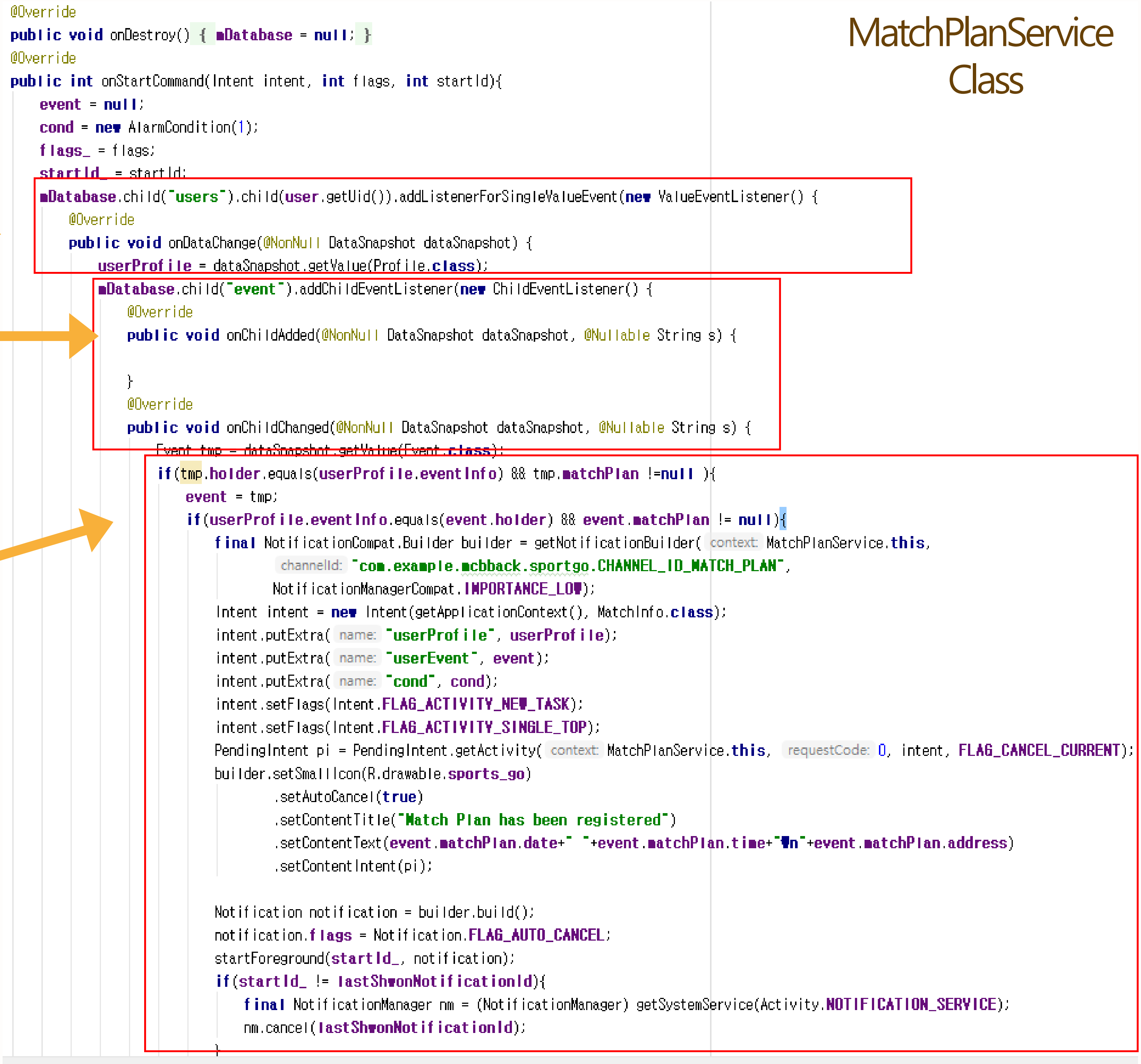
### 10. Alarm Services - Match Plan Alarm

MatchPlanService Class

Get the user profile

Add event database listener that detect changes of event

When the listener detects the changes of events, check that the changed event is the event that user taken and changed event has the match plan. If true, the service send notification to the user device.

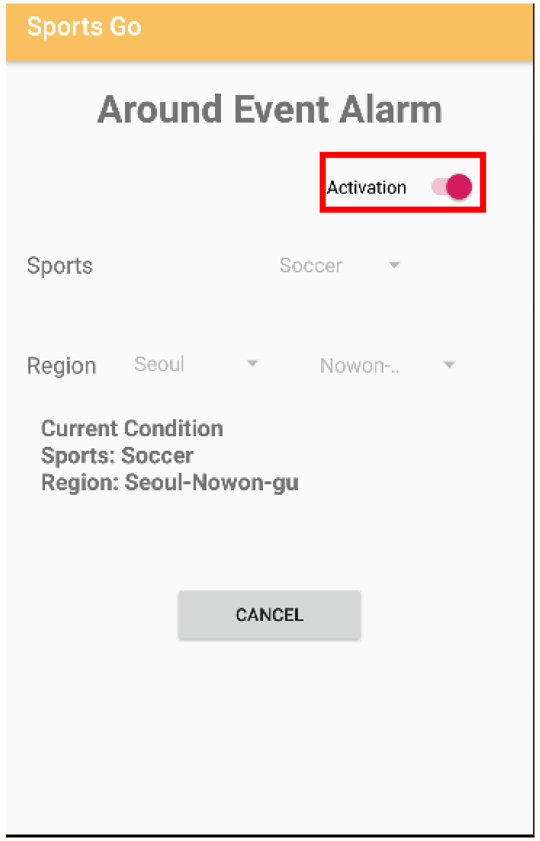


```
@Override
public void onDestroy() { mDatabase = null; }
@Override
public int onStartCommand(Intent intent, int flags, int startId){
    event = null;
    cond = new AlarmCondition(1);
    flags_ = flags;
    startId_ = startId;
    mDatabase.child("users").child(user.getUid()).addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            userProfile = dataSnapshot.getValue(Profile.class);
            mDatabase.child("event").addChildEventListener(new ChildEventListener() {
                @Override
                public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {

                }
                @Override
                public void onChildChanged(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {
                    Event tmp = dataSnapshot.getValue(Event.class);
                    if(tmp.holder.equals(userProfile.eventInfo) && tmp.matchPlan !=null ){
                        event = tmp;
                        if(userProfile.eventInfo.equals(event.holder) && event.matchPlan != null){
                            final NotificationCompat.Builder builder = getNotificationBuilder( context: MatchPlanService.this,
                                     channelId: "com.example.mcbback.sportgo.CHANNEL_ID_MATCH_PLAN",
                                    NotificationManagerCompat.IMPORTANCE_LOW);
                            Intent intent = new Intent(getApplicationContext(), MatchInfo.class);
                            intent.putExtra( name: "userProfile", userProfile);
                            intent.putExtra( name: "userEvent", event);
                            intent.putExtra( name: "cond", cond);
                            intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                            intent.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
                            PendingIntent pi = PendingIntent.getActivity( context: MatchPlanService.this,  requestCode: 0, intent, FLAG_CANCEL_CURRENT);
                            builder.setSmallIcon(R.drawable.sports_go)
                                    .setAutoCancel(true)
                                    .setContentTitle("Match Plan has been registered")
                                    .setContentText(event.matchPlan.date+" "+event.matchPlan.time+"\n"+event.matchPlan.address)
                                    .setContentIntent(pi);

                            Notification notification = builder.build();
                            notification.flags = Notification.FLAG_AUTO_CANCEL;
                            startForeground(startId_, notification);
                            if(startId_ != lastShwonNotificationId){
                                final NotificationManager nm = (NotificationManager) getSystemService(Activity.NOTIFICATION_SERVICE);
                                nm.cancel(lastShwonNotificationId);
                            }
```

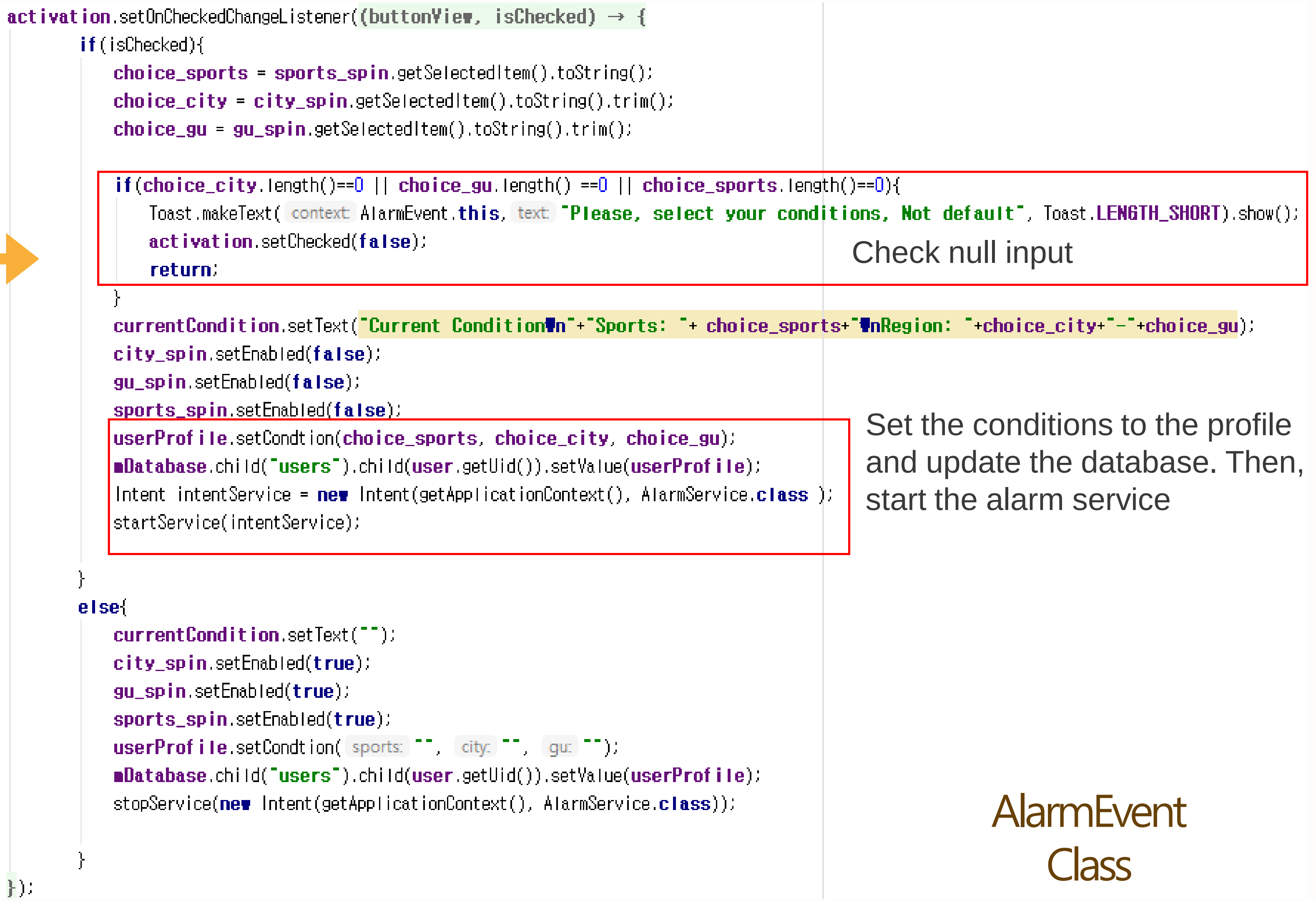
03 Main Functions – Source code

## Alarm service

### 10. Alarm Services – Around Event Alarm



Current Condition
Sports: Soccer
Region: Seoul-Nowon-gu



```
activation.setOnCheckedChangeListener((buttonView, isChecked) → {
        if(isChecked){
            choice_sports = sports_spin.getSelectedItem().toString();
            choice_city = city_spin.getSelectedItem().toString().trim();
            choice_gu = gu_spin.getSelectedItem().toString().trim();

            if(choice_city.length()==0 || choice_gu.length() ==0 || choice_sports.length()==0){
                Toast.makeText( context: AlarmEvent.this, text: "Please, select your conditions, Not default", Toast.LENGTH_SHORT).show();
                activation.setChecked(false);
                return;
            }
            currentCondition.setText("Current Condition₩n"+"Sports: "+ choice_sports+"₩nRegion: "+choice_city+"-"+choice_gu);
            city_spin.setEnabled(false);
            gu_spin.setEnabled(false);
            sports_spin.setEnabled(false);
            userProfile.setCondtion(choice_sports, choice_city, choice_gu);
            mDatabase.child("users").child(user.getUid()).setValue(userProfile);
            Intent intentService = new Intent(getApplicationContext(), AlarmService.class );
            startService(intentService);

        }
        else{
            currentCondition.setText("");
            city_spin.setEnabled(true);
            gu_spin.setEnabled(true);
            sports_spin.setEnabled(true);
            userProfile.setCondtion( sports: "", city: "", gu: "");
            mDatabase.child("users").child(user.getUid()).setValue(userProfile);
            stopService(new Intent(getApplicationContext(), AlarmService.class));

        }
});
```

Check null input

Set the conditions to the profile and update the database. Then, start the alarm service

AlarmEvent Class

## Alarm service

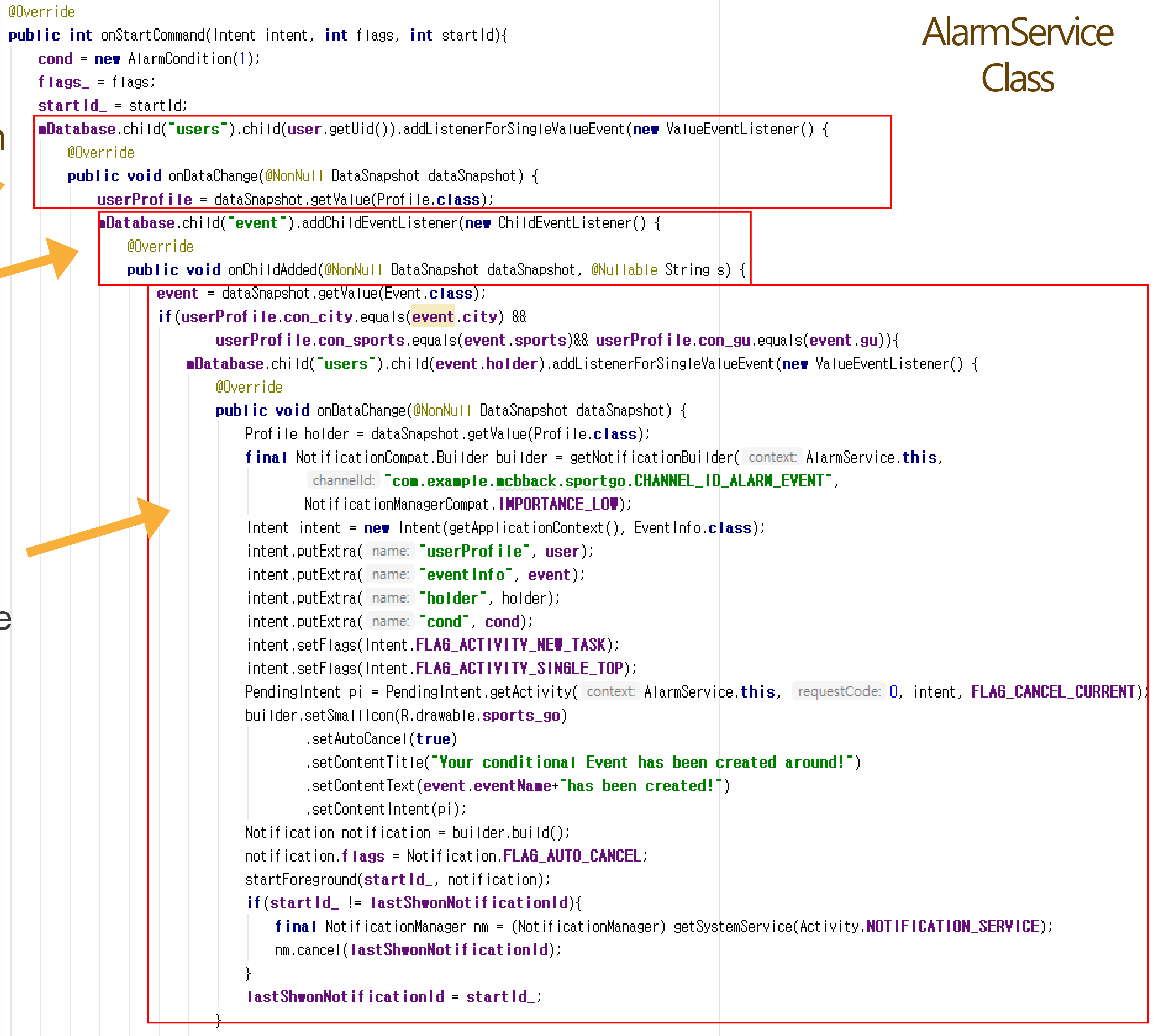
### 10. Alarm Services – Around Event Alarm

AlarmService Class

Get the user profile

Add event database listener that detect new added event

When the listener detects the new added events, check that the new event's feature is same as the user's alarm conditions. If true, the service send notification to the user device.



```
@Override
public int onStartCommand(Intent intent, int flags, int startId){
    cond = new AlarmCondition(1);
    flags_ = flags;
    startId_ = startId;
    mDatabase.child("users").child(user.getUid()).addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            userProfile = dataSnapshot.getValue(Profile.class);
            mDatabase.child("event").addChildEventListener(new ChildEventListener() {
                @Override
                public void onChildAdded(@NonNull DataSnapshot dataSnapshot, @Nullable String s) {
                    event = dataSnapshot.getValue(Event.class);
                    if(userProfile.con_city.equals(event.city) &&
                            userProfile.con_sports.equals(event.sports)&& userProfile.con_gu.equals(event.gu)){
                        mDatabase.child("users").child(event.holder).addListenerForSingleValueEvent(new ValueEventListener() {
                            @Override
                            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                                Profile holder = dataSnapshot.getValue(Profile.class);
                                final NotificationCompat.Builder builder = getNotificationBuilder( context: AlarmService.this,
                                         channelId: "com.example.mcbback.sportgo.CHANNEL_ID_ALARM_EVENT",
                                        NotificationManagerCompat.IMPORTANCE_LOW);
                                Intent intent = new Intent(getApplicationContext(), EventInfo.class);
                                intent.putExtra( name: "userProfile", user);
                                intent.putExtra( name: "eventInfo", event);
                                intent.putExtra( name: "holder", holder);
                                intent.putExtra( name: "cond", cond);
                                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                                intent.setFlags(Intent.FLAG_ACTIVITY_SINGLE_TOP);
                                PendingIntent pi = PendingIntent.getActivity( context: AlarmService.this,  requestCode: 0, intent, FLAG_CANCEL_CURRENT);
                                builder.setSmallIcon(R.drawable.sports_go)
                                        .setAutoCancel(true)
                                        .setContentTitle("Your conditional Event has been created around!")
                                        .setContentText(event.eventName+"has been created!")
                                        .setContentIntent(pi);
                                Notification notification = builder.build();
                                notification.flags = Notification.FLAG_AUTO_CANCEL;
                                startForeground(startId_, notification);
                                if(startId_ != lastShwonNotificationId){
                                    final NotificationManager nm = (NotificationManager) getSystemService(Activity.NOTIFICATION_SERVICE);
                                    nm.cancel(lastShwonNotificationId);
                                }
                                lastShwonNotificationId = startId_;
                            }
```

## 004 Conclusion

- Improvement
- Trough the project

## Improvement

### 1 Layout

- Make the UI more comfortable
- Change logo color to match the UI color

### 2 Functions

- Modify chatting function with supporting user thumbnail
- Add more sports facility database
- Expand Available Areas

## Trough the project

### 1 Need Detailed information in function planning

- We plan the function in a big framework.
- It was difficult to break down the functions in detail

### 2 Need to study source code management tools

- We use Github and source tree
- There were some errors at first because I didn't know how to use them well.
- It took a long time to integrate each functions into one program

## Q & A

Thank You