**모바일 앱 프로그래밍**

**1차 과제 보고서**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 제출일 | 2020.05.05 |  | 전공 | 소프트웨어공학과 |
| 과목 | 모바일 앱 프로그래밍 |  | 담당교수 | 김순태 교수님 |
| **9조** 이정훈 김보민 | | | | |



1. **프로젝트 개요**

**A simple game 기본 어플리케이션에 여러가지 개선 기능을 넣어 게임을 개선하는 프로젝트 입니다.**

1. **팀원**



**-팀원 1**

**학과 : 소프트웨어공학과**

**학번 : 201413311**

**학년 : 3학년**

**이름 : 이정훈**

**- 팀원 2**

**학과 : 소프트웨어공학과**

**학번 : 201510742**

**학년 : 4학년**

**이름 : 김보민**

1. **개선 아이디어**

: 파이어베이스의 구현을 해보고 싶기도 하고 마이페이지를 로그인 후 구현하면 좋을 거 같아 파이어베이스 로그인 기능으로 아이디어를 변경했습니다.

**변경 전**

|  |  |
| --- | --- |
| 개선아이디어 | 난이도 |
| 효과음 및 배경음 기능 (Sound class 사용) | 1.5 |
| 인 게임 화면에 다시하기 버튼 추가-다시하기 버튼 추가 | 0.5 |
| 문제 랜덤으로 출력하기-게임을 실행할 때마다 랜덤으로 숫자를 db에 입력시켜 항상 다른 문제를 출력. | 0.5 |
| 마이페이지(본인 최고기록, 평균점수) | 1 |
| TTS를 사용하여 문제를 내면, STT를 사용하여 답을 말해서 문제 풀기 | 2.5 |
| 틀린문제 확인하는 기능 추가-게임 진행 도중 틀린문제가 발생했을때 db에 저장시켜 게임이 끝난 후 확인가능 | 2.5 |
| 난이도에 따라 다른 문제가 나오게 한다. | 필수 |
| 랭킹 기능을 추가한다. | 필수 |
| MainPage를 만든다. | 필수 |
| 난이도 총합 | **8.5** |

**변경 후**

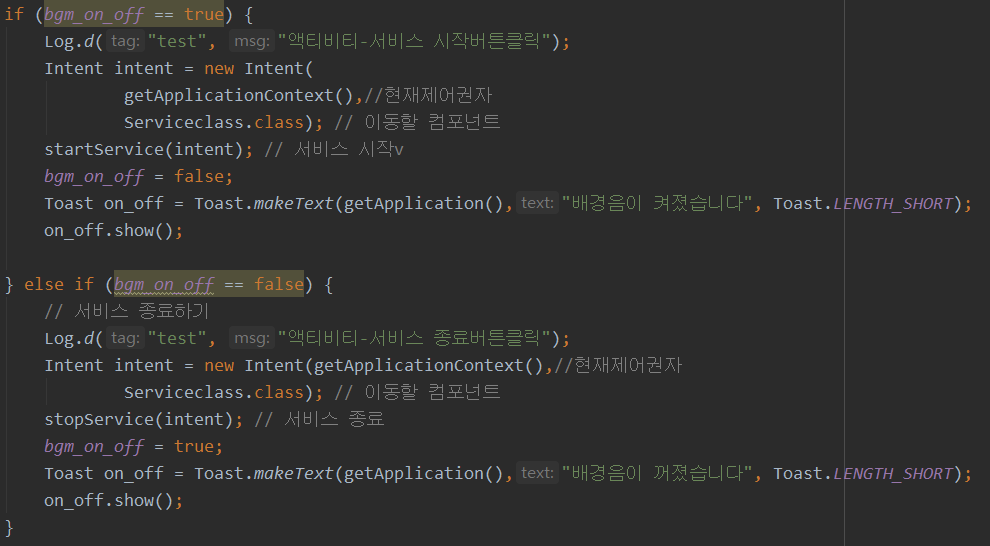
|  |  |
| --- | --- |
| 개선아이디어 | 난이도 |
| 효과음 및 배경음 기능 (Sound class 사용) | 1.5 |
| 인 게임 화면에 다시하기 버튼 추가-다시하기 버튼 추가 | 0.5 |
| 문제 랜덤으로 출력하기-게임을 실행할 때마다 랜덤으로 숫자를 db에 입력시켜 항상 다른 문제를 출력. | 0.5 |
| 로그인 시스템(파이어 베이스 데이터베이스를 이용한 로그인 시스템 구현) | 3.5 |
| TTS를 사용하여 문제를 내면, STT를 사용하여 답을 말해서 문제 풀기 | 2.5 |
| 틀린 문제 확인하는 기능 추가-게임 진행 도중 틀린 문제가 발생했을 때 DB에 저장시켜 게임이 끝난 후 확인가능 | 2.5 |
| 난이도에 따라 다른 문제가 나오게 한다. | 필수 |
| 랭킹 기능을 추가한다. | 필수 |
| MainPage를 만든다. | 필수 |
| 난이도 총합 | **11.5** |

1. **필수 및 개선 아이디어 설명**
2. **배경음 기능**
   1. **구현을 위한 주요 소스코드 및 설명**

**<ServiceClass.java의 코드>**

**<설명>**

Service를 extends 하는 클래스를 만든 뒤 onCreate()안에 재생할 음악파일을 지정해준다. onStartCommand()는 StatrService()가 호출되면 같이 호출되는 메소드로써 return Service.START\_NOT\_STICKY 로 게임 종료 시에도 음악이 재생되는 것을 방지한다.

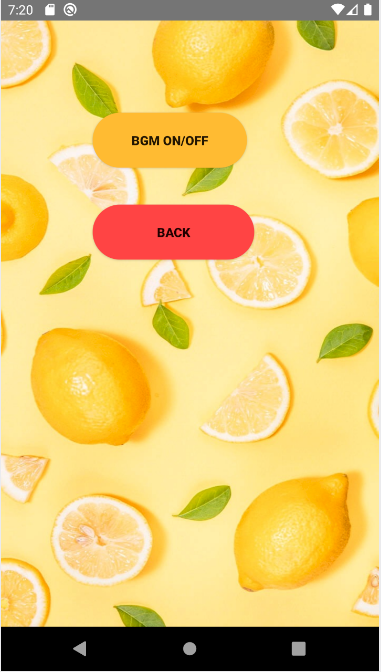


**<SettingActivity.java의 코드>**

**<설명>**

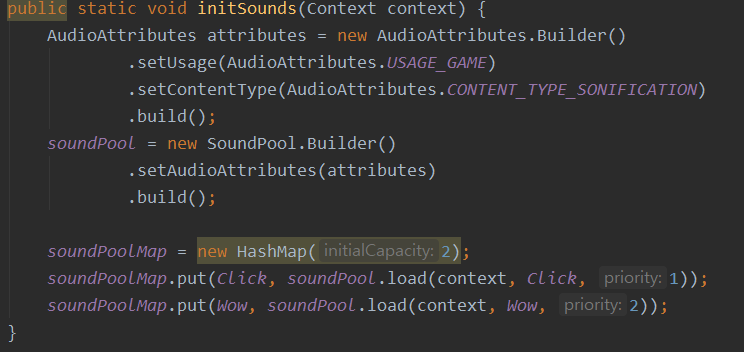
버튼 클릭 시 BGM을 On/Off 하도록 if 문을 사용하였다.

* 1. **스크린 캡쳐**



**<설명>**

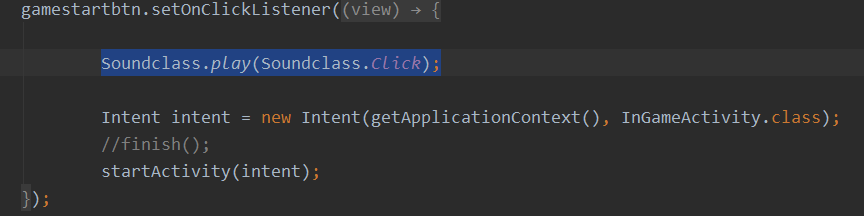
Activity\_setting.xml- BGM on/off 클릭 시 BGM이 on/off 된다. 처음 설정 BGM은 Off상태이다.

1. **효과음 기능**
   1. **구현을 위한 주요 소스코드 및 설명**

**<Soundclass.java의 코드>**

**<설명>**

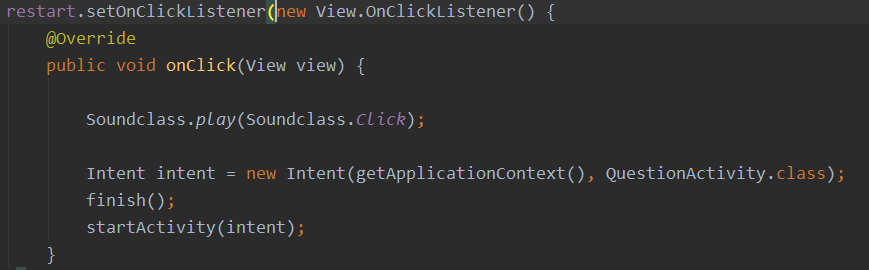
효과음 구현을 위해 사용되는 것에는 SoundPool 과 MediaPlayer가 있다. SoundPool은 알림 효과, 게임효과 등의 짧은 사운드클립에 적합하고 MediaPlayer는 노래와 같이 더 큰 사운드 파일을 재생하는데 도움이 된다. initSounds에 추가할 효과음파일을 추가한다.



**<MainScreenActivity.java의 코드>**

**<설명>**

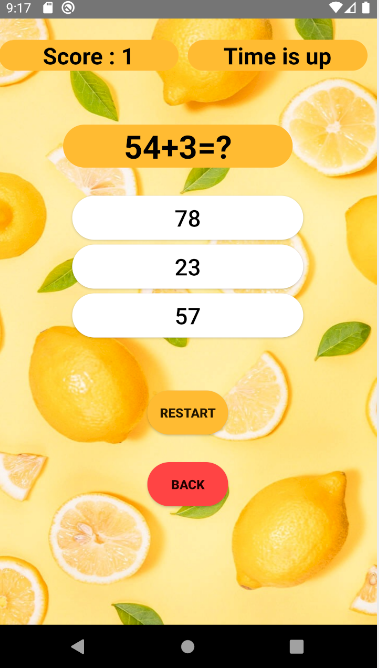
클릭 이벤트 발생 시마다 play메소드로 효과음이 나오도록 setOnClickListner() 안에 Paly()메소드를 추가함으로써 구현하였다.

1. **다시하기 기능**
   1. **구현을 위한 주요 소스코드 및 설명**

**<QuestionActivity.java의 코드>**

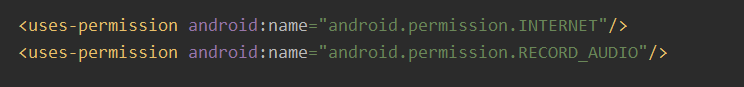
**<설명>**

플레이 게임 화면에 restart 버튼을 만들고 startActivity로써 게임을 다시 불러온다

* 1. **스크린 캡쳐**

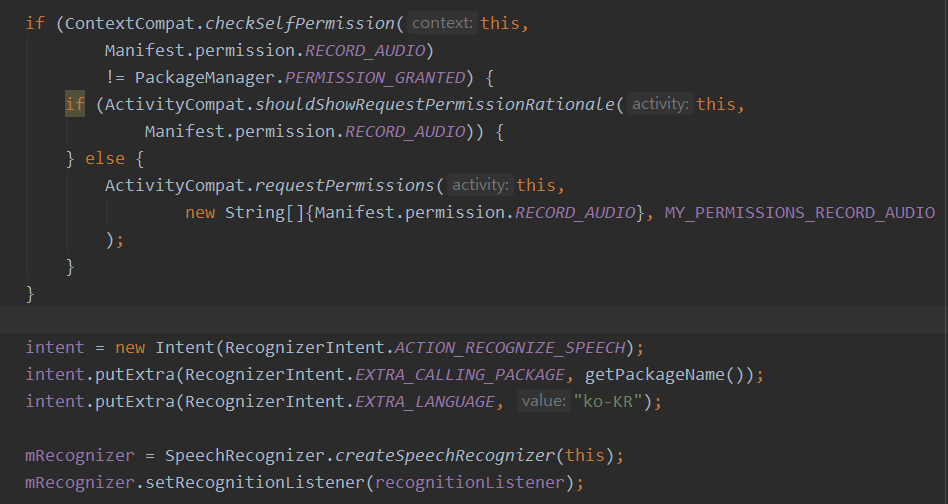
**<설명>**

RESTART 버튼 클릭 시 해당 난이도 게임이 재시작 된다.

1. **TTS STT 음성 문제 기능**
   1. **구현을 위한 주요 소스코드 및 설명**

**<manifests의 코드>**

**<설명>**

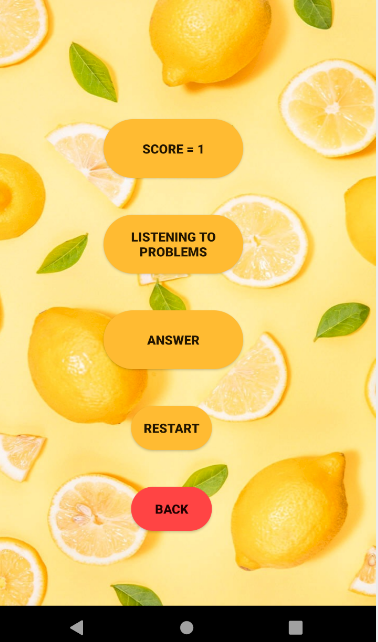
TTS, STT와 관련된 권한을 설정해주는 코드이다

**<TtsStt.java의 코드>**

**<설명>**

구글의 SpeechRecognizer API는 한번 음성을 인식한 후, 끝나 버리기 때문에 계속 재시작을 해주는 식으로 구현해야 한다. 따라서 STT를 위한 버튼을 만들고 버튼을 클릭할 때에만 음성을 인식하도록 구현하였다. TTS를 위한 변수 tts를 선언하였고, setLanguage()를 통해 한국어로 초기 설정을 한다. ttsbtn 클릭 시 해당 textView의 string을 얻어 한국어로 음성이 출력 된다.

* 1. **스크린 캡쳐**



**<설명>**

Listen to problems 버튼을 누르면 문제를 들을 수 있으며, Answer버튼을 누르고 정답을 말하면 정답을 인식하여 간단한 메시지를 통해 정답 여부를 확인할 수 있다.

1. **Main Page 생성**
   1. **구현을 위한 주요 소스코드 및 설명**

**<activity\_main\_screen.xml의 코드>**

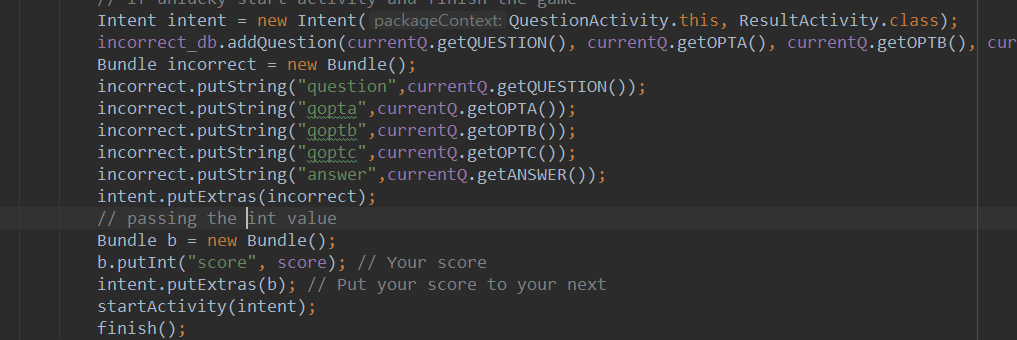
**<설명>**

Linearayout으로 각각의 기능에 맞는 버튼을 추가하여 구성하였다.

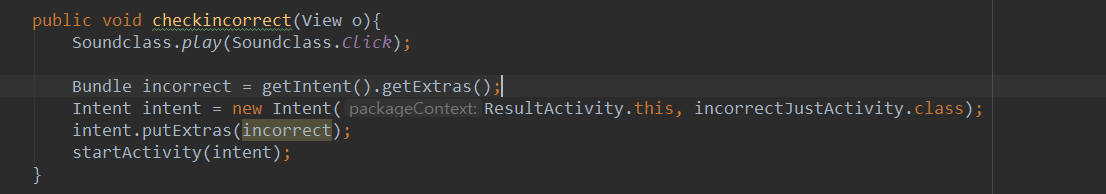
* 1. **스크린 캡쳐**



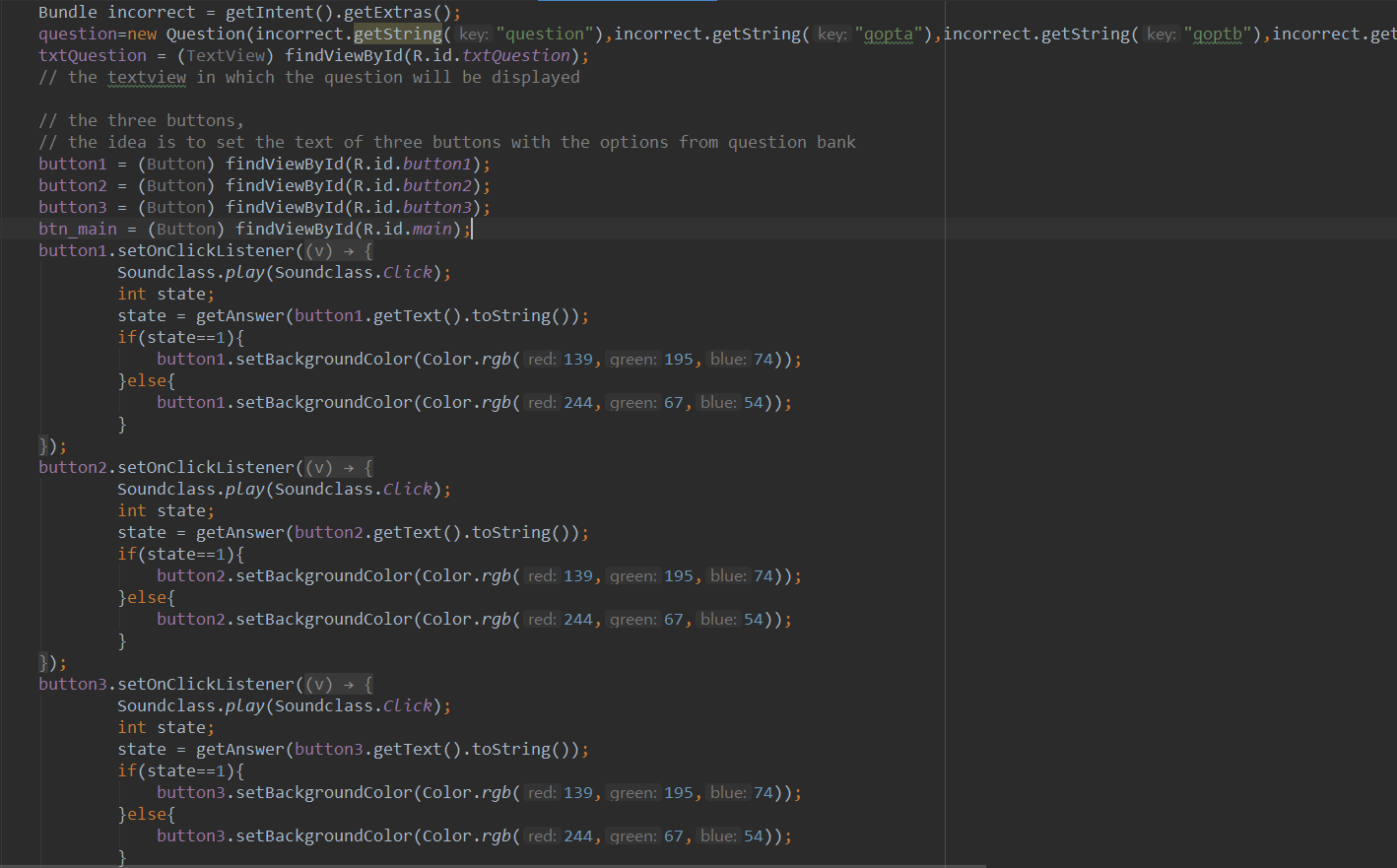
1. **틀린 문제 확인 기능**
   1. **구현을 위한 주요 소스코드 및 설명**
      1. **틀린문제 바로 확인하기**



**<QustionAcitivity.java 의 코드>**



**<ResultAcitivity.java 의 코드>**



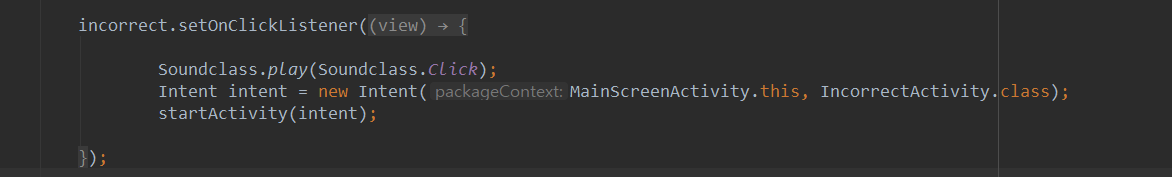
**<IncorrectJustActivity.java의 코드>**

**<설명>**

QuestionActivity에서 틀린문제가 발생하는 즉시 Bundle에 문제와 정답과 보기들을 받아 ResultActivity로 넘겨줍니다. 그다음 ResultActivity에서 틀린문제 확인하기 버튼을 눌렀을 때 받아 온 정보들을 다시 IncorrectJustActivity.java로 넘겨주어 방금틀린문제를 바로 확인할 수 있게 구현하였습니다.

틀린문제화면에서는 정답이라 생각되는 것을 눌러보고 정답일 시에 초록색으로 버튼이 바뀌고 오답일 시 빨간색으로 버튼이 바뀌어 정답을 확인할 수 있게 구현하였습니다.

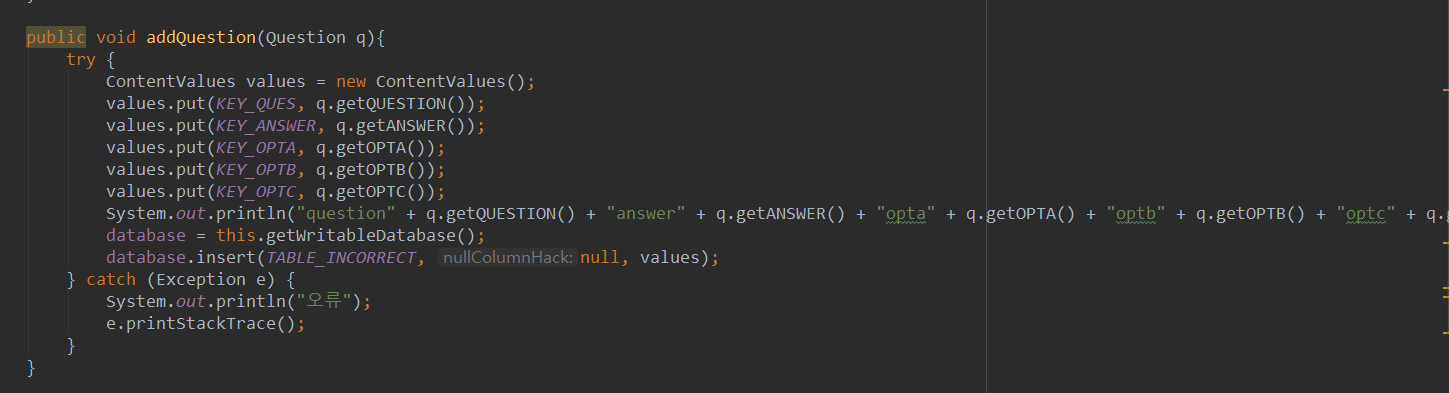
* + 1. **틀린문제 모아보기**



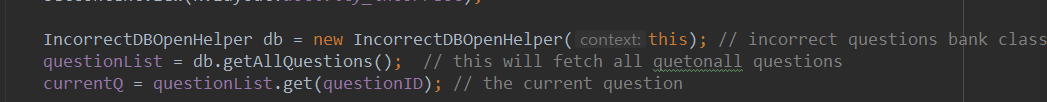
**<MainScreenActivity.java의 코드>**

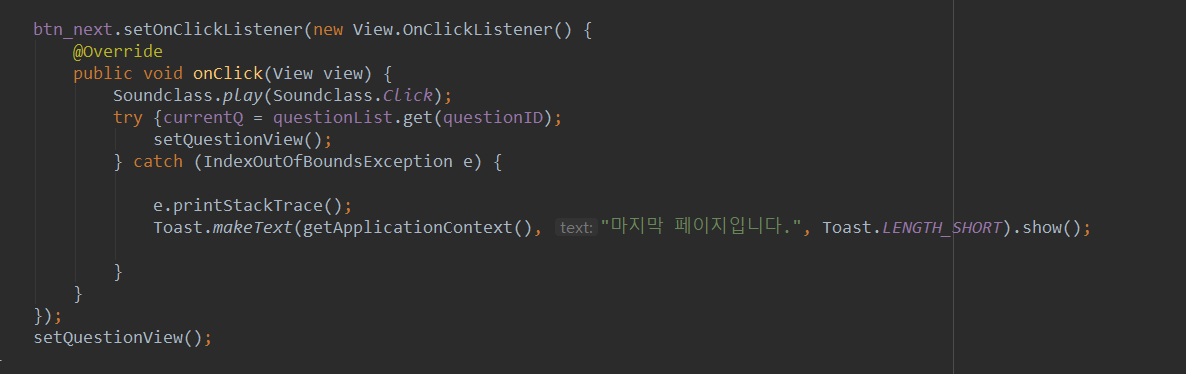


**<QuestionActivity.java의 코드>**



**<IncorrectDBOpenHelper.java의 코드>**





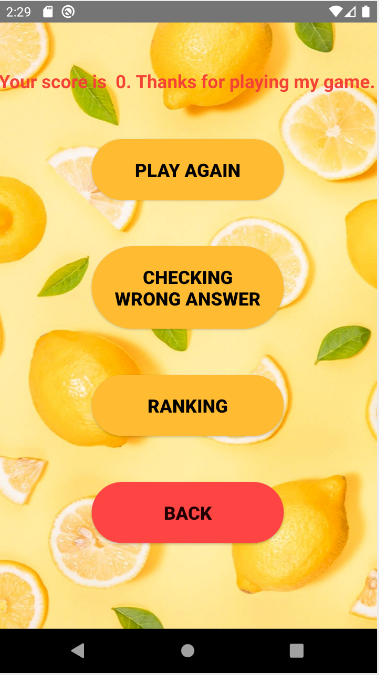
**<IncorrectActivity.java의 코드>**

**<설명>**

QuestionActivity에서 틀린문제가 발생하면 IncorrectDBOpenHelper클래스를 이용해서 incorrect DB에 틀린문제를 저장합니다. 메인에서 틀린문제확인하기 버튼을 클릭하면 IncorrectActivity로 넘어가 저장된 틀린문제들을 볼 수 있고 틀린문제를 바로 확인할때와 마찬가지고 보기를 눌렀을 때 틀린답이면 빨간색으로 정답을 골랐을때는 초록색으로 버튼이 바뀌어 정답을 확인해볼 수 있습니다. 그리고 마지막페이지에서 다음문제를 확인하는 버튼을 눌렀을때는 예외처리를 통해 마지막 페이지라는 토스트 메시지를 띄워줄 수 있게 구현하였습니다.

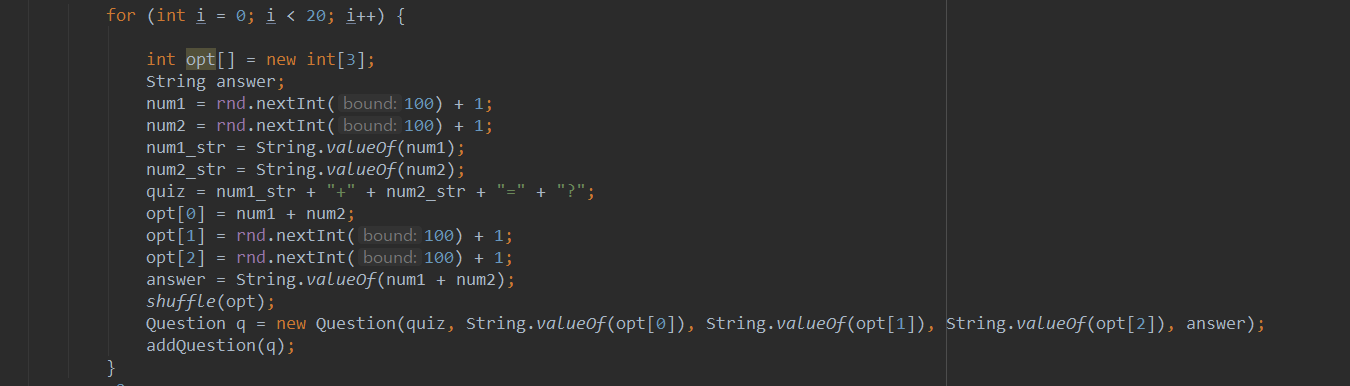
* 1. **스크린 캡쳐** 
     1. **틀린문제 바로 확인하기**

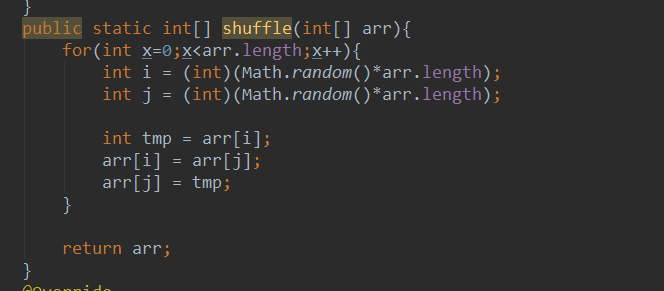
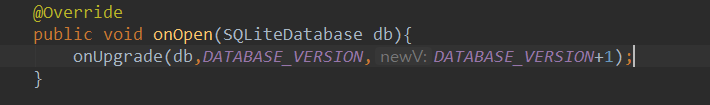




* + 1. **틀린문제 모아보기**



1. **문제 랜덤 출제**
   1. **구현을 위한 주요 소스코드 및 설명**



**<QuizDBOpenHelper클래스의 코드>**

**<설명>**

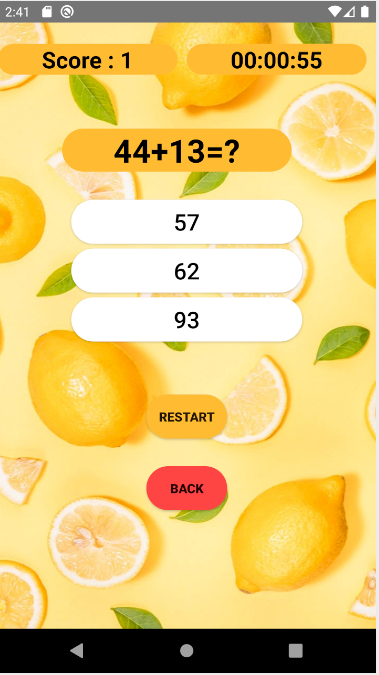
Random을 사용해서 문제와 정답외의 보기를 random으로 보여줄 수 있게 설정을 해줬습니다.

보기에 계속 같은 위치에 답이 나오는 것을 해결하기 위해 shuffle 함수를 구현하여서 문제의 option들이 섞여서 랜덤으로 보기가 출력 될 수 있게 설정해주었습니다. 게임을 실현될때마다 새로운 문제를 출력해야해서 onUpgrade함수를 이용해서 항상 새로운 DB를 만들어 새로운 문제들만 출력할 수 있게 해주었습니다.

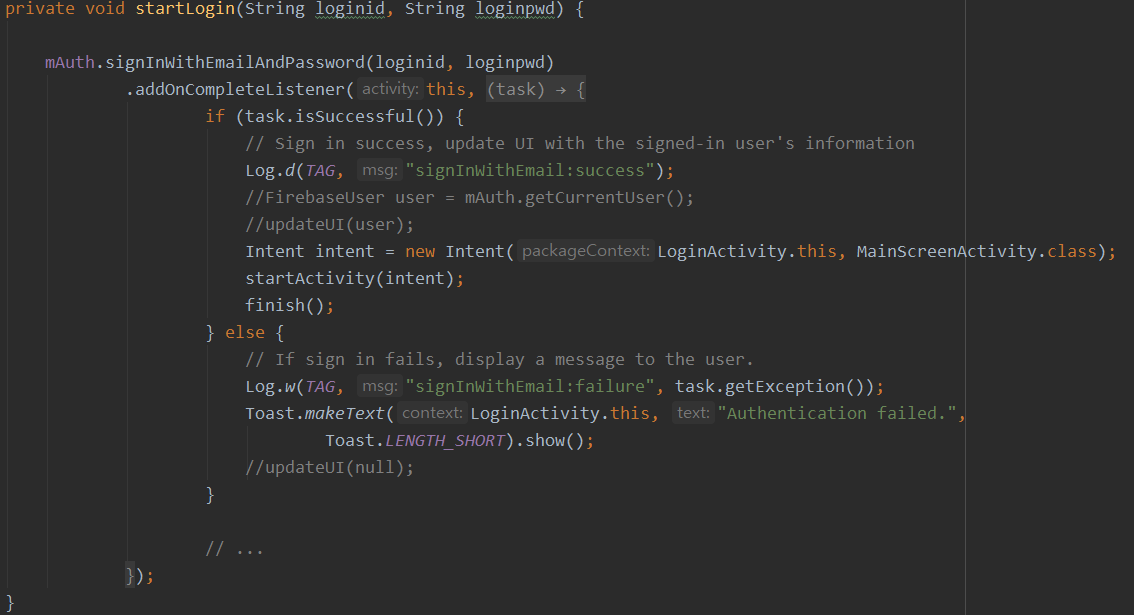
* 1. **스크린 캡쳐**



**<랜덤문제 1>**



**<랜덤문제2>**

1. **로그인/회원가입 기능**
   1. **구현을 위한 주요 소스코드 및 설명**
      1. **로그인**

**<LoginActivity.java의 코드>**

**<설명>**

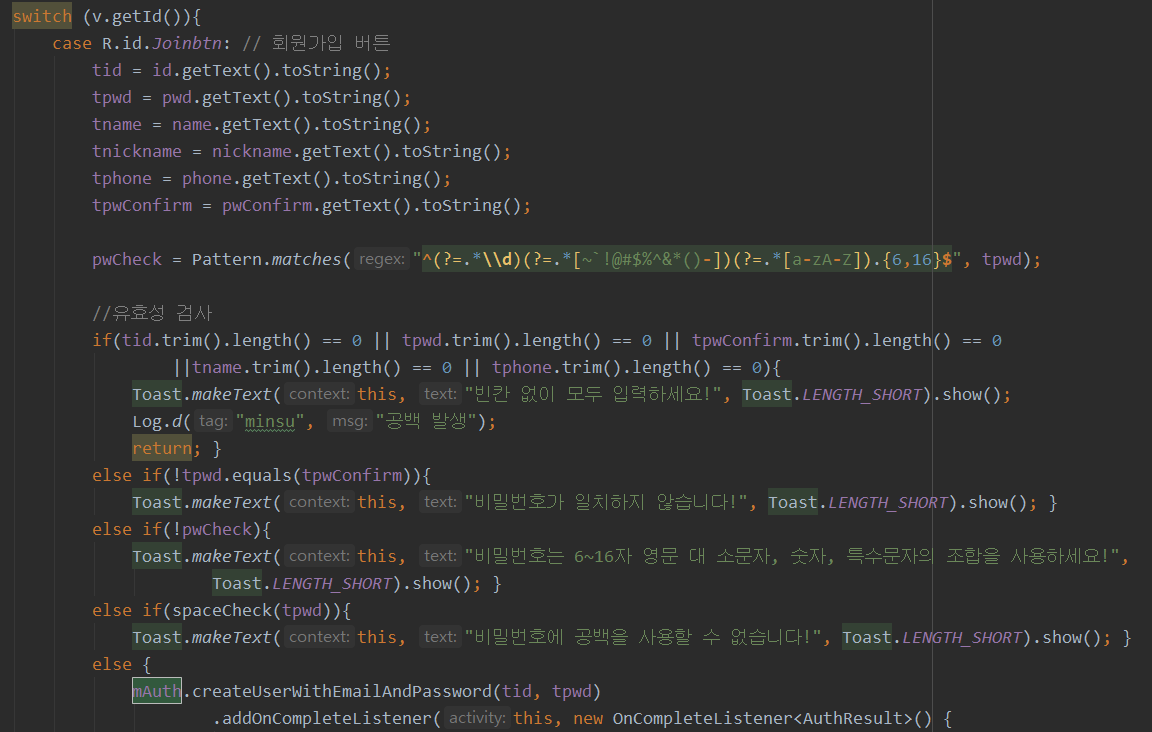
loginid와 loginpwd를 입력 값으로 받아 firebase에 authentication과정을 거쳐 인증이 제대로 이루어지면 다음창으로 넘어가고 이루어지지 않으면 “Authentication failed”를 띄어 인증이 실패하였음을 알려줍니다.

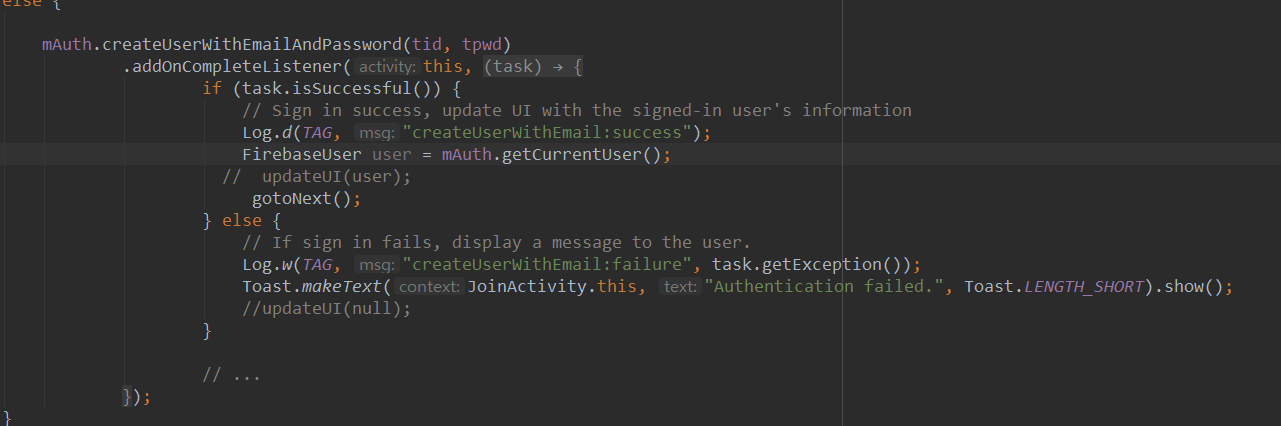
1. **B 스크린 캡쳐**

**<로그인 화면>**

**<로그인 인증실패시>**

**b. 회원가입**



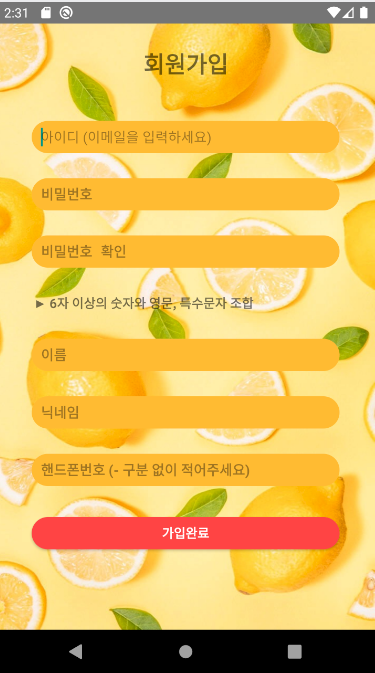
**<JoinActivity.java의 코드>**



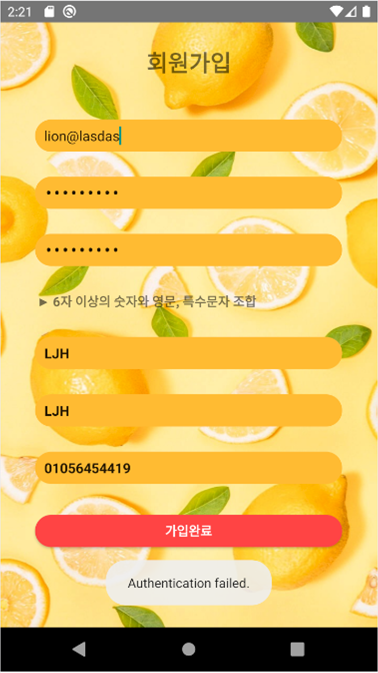
**<firebase의 회원가입 인증확인>**

**<설명>**

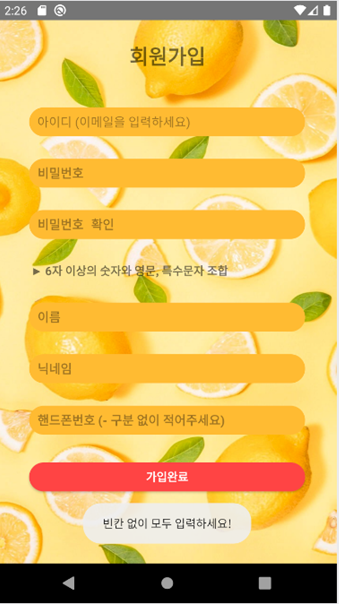
Id, pwd, name, nickname, email, phone을 입력 받아 firebase에 저장한 뒤 새로운 아이디를 만듭니다. 자체에 유효성 검사를 넣어서 원하는 입력값들이 칸들에 입력이 되어있지 않거나, 비밀번호와 비밀번호 확인이 일치하지 않을 때 toast메시지를 만들어 제대로 된 입력을 요구하고 비밀번호는 6~16자의 영문 대소문자, 숫자, 특수문자의 조합을 사용하여야 합니다. 유효성 검사를 통과하면 firebase에 인증을 통해 회원가입이 이루어집니다.

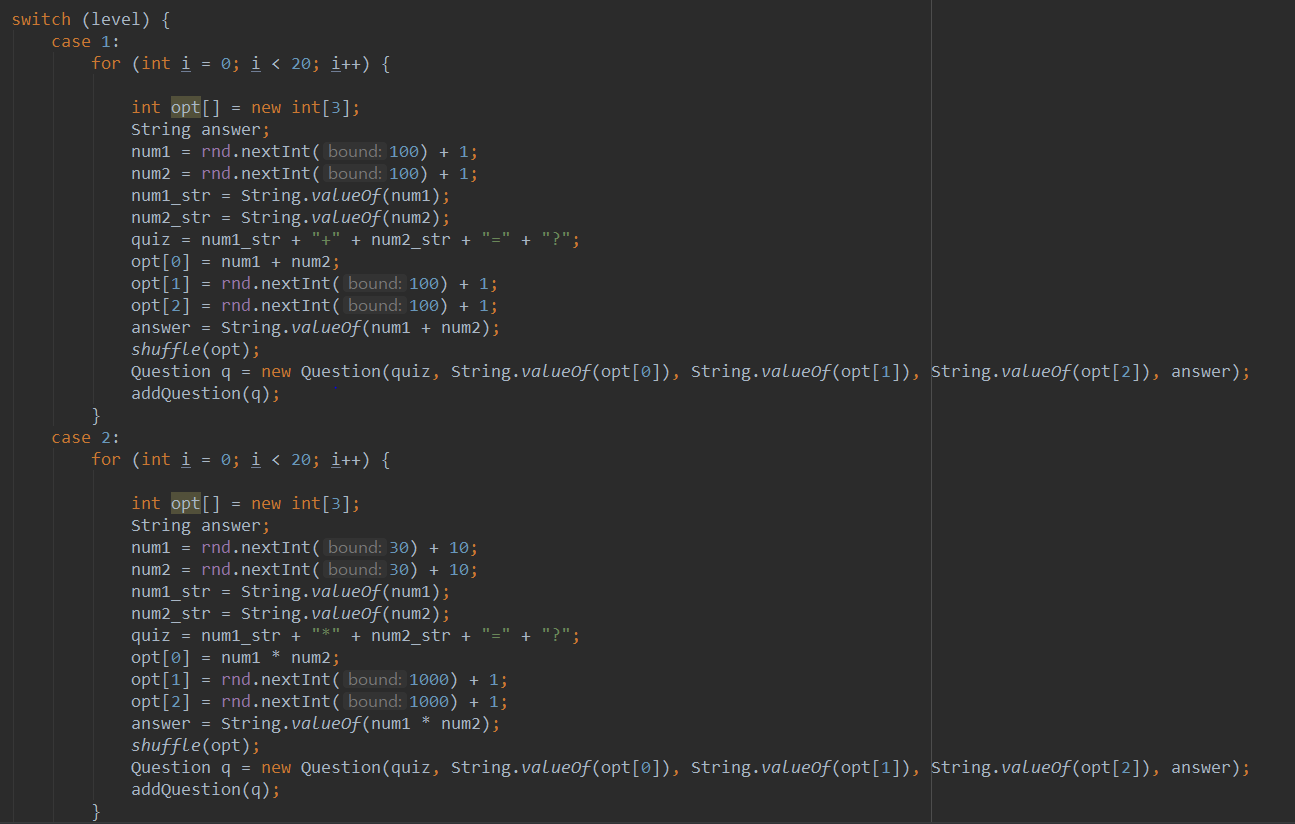
1. **B 스크린 캡쳐**

**<회원가입 화면>**

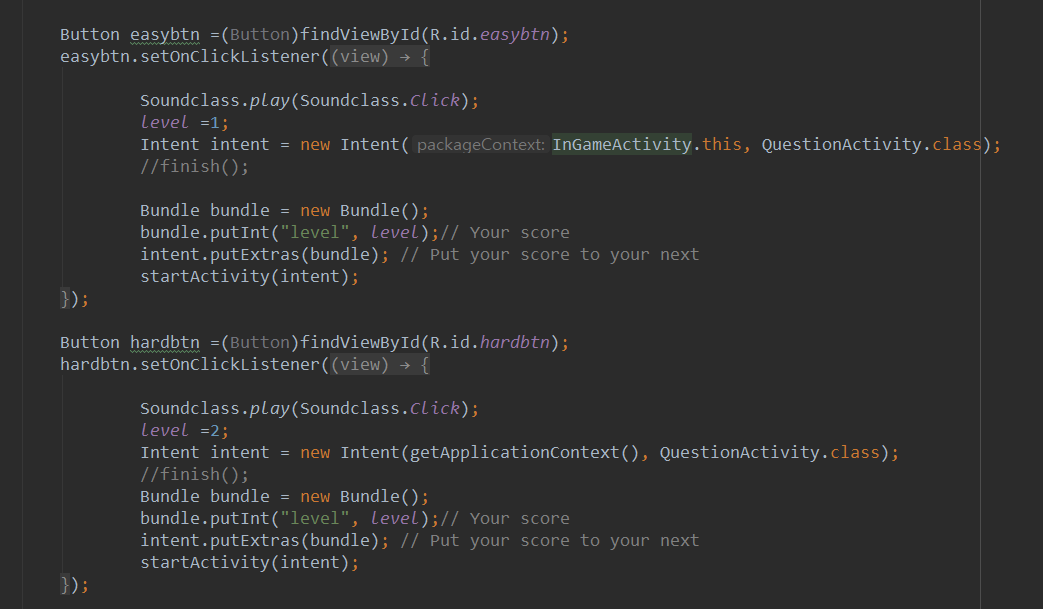
**<이메일 형식이 올바르지 않을시>**

**<비밀번호가 일치하지 않을시 화면>**

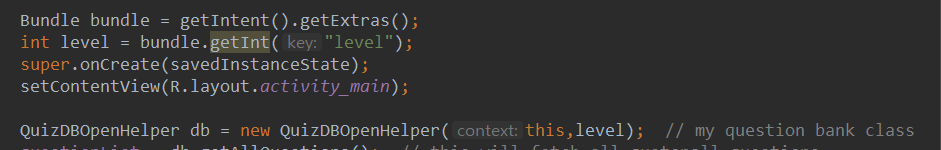
**<회원가입시 공백입력 화면>**

1. **난이도 추가** 
   1. **구현을 위한 주요 소스코드 및 설명**

**<quizDBOpenHelper.java의 코드>**



**<InGameActivity.java의 코드>**

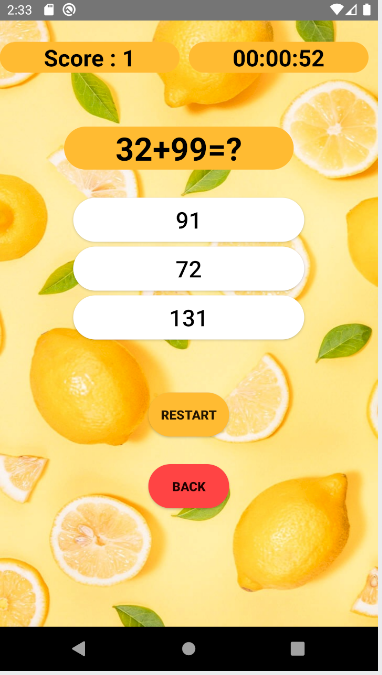


**<QuestionActivity.java의 코드>**

**<설명>**

위에서 랜덤으로 문제를 만들었던 것을 참고해서 hard버전은 곱하기로 문제를 내기로 하였습니다. 다만, 곱하기이다보니 정답이 너무 커져서 오히려 정답을 고르기 쉬워지는 현상이 발생하여 문제의 범위를 30까지로 설정하고 보기는 100의자리까지로 설정해서 난이도를 설정했습니다. Easybtn을 누르면 level이 1로 설정되고 hardbtn을 누르면 level이 2로 설정되어 level을 Intent로 넘겨주어 QuestionActivity 클래스에서 Intent로 level을 받아주고 클래스에서 생성자를 생성할 때 level을 함께 넘겨주어 난이도를 설정할 수 있게 구현하였습니다.

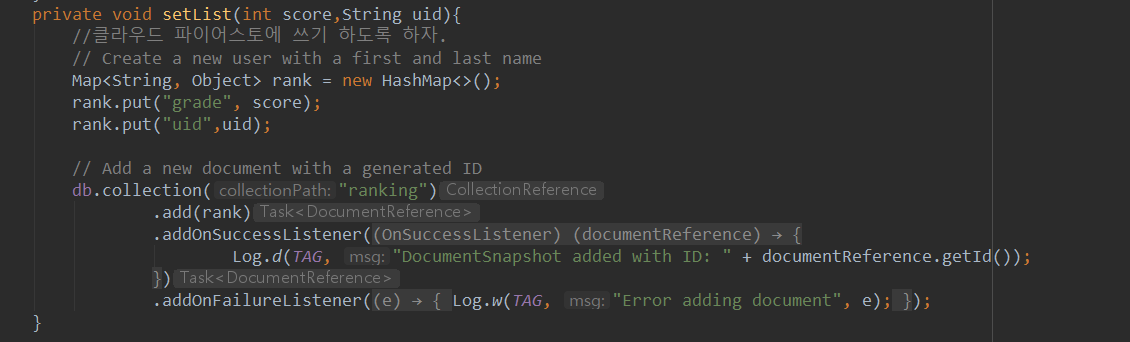
* 1. **스크린 캡쳐**



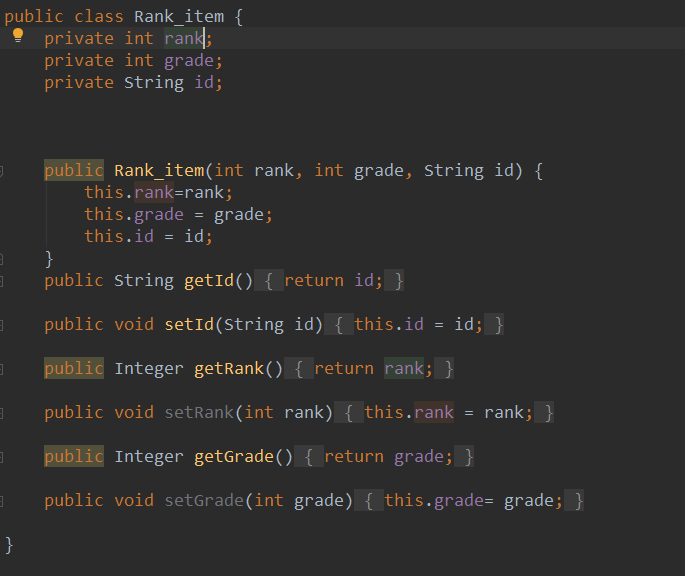
**<이지모드>**

**<하드모드>**

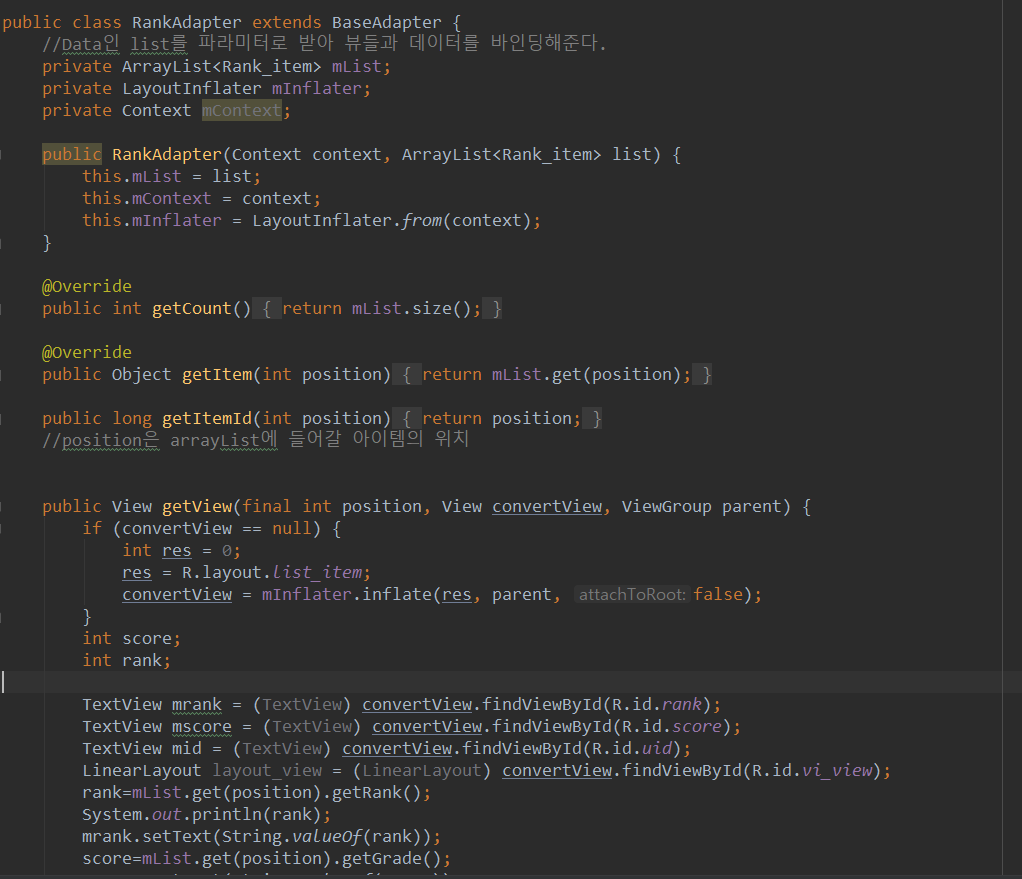
1. **랭킹기능 구현**
   1. **구현을 위한 주요 소스코드 및 설명**



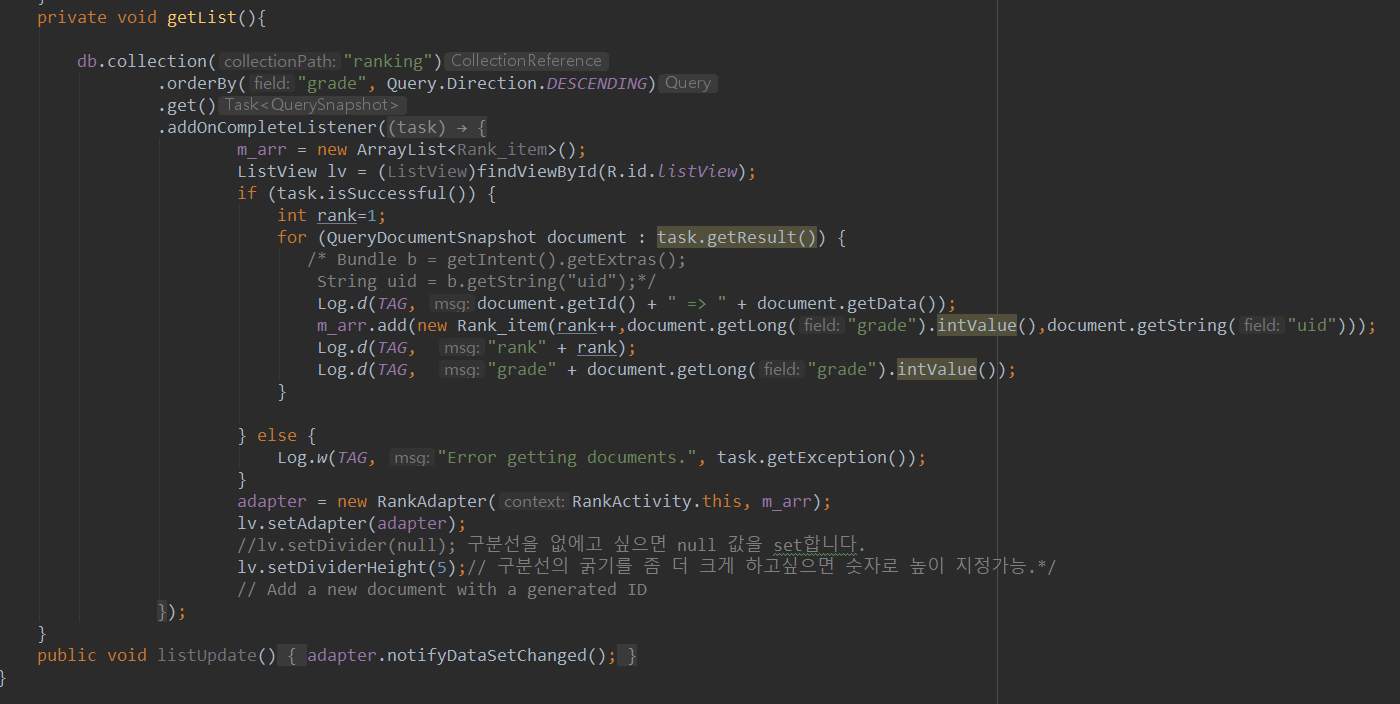
**<ResultActivity.java의 구현>**



**<Rank\_item.java의 구현>**



**<RankAdapter.java의 구현>**



**<RankActivity.java의 구현>**

**<설명>**

파이어베이스 클라우드 파이어스토어로 랭킹을 구현하였습니다. ResultActivity에서 점수와 userid를 받아와서 게임이 끝나면 결과창에서 파이어스토어에 저장해주었습니다.

Rank\_item에서 리스트에서 보여줄 점수와 uid를 포함한 객체를 생성해주었고 RankAdapter를 이용해서 Data인 리스트를 파라미터로 받아 뷰들과 데이터를 바인딩해주었습니다.

그리고 RankActivity.java에서는 파이어스토어의 값들을 받아내 출력해줄 수 있는 getList()를 구현하였습니다. 디비에는 아이디와 점수가 저장되고 점수에 따른 sorting을 오름차순으로 설정해 점수순으로 보여줄 수 있게끔 설정하였습니다.

* 1. **스크린 캡쳐**



**<랭킹화면>**

**<firebase cloud Firestore>**

1. **향후 개선 아이디어**
2. **아이디 / 비밀번호 찾기**

**아이디와 비밀번호 찾기의 UI는 만들어 놓은 상태이다. Firestore를 이용하여 구현**

1. **자동로그인 기능**

**현재 게임을 시작할때마다 로그인을 해줘야 하는 상황이다 자동로그인 기능을 추가하여 같은기기에서는 로그인을 다시 할 필요가 없는 기능을 추가**

1. **마이페이지 기능**

**회원가입시 작성하는 내정보와, 점수의 평균 최고점수와 최저점수등을 알 수 있는 마이페이지 기능추가**

1. **스테이지 별 랭킹 기능**

**현제 랭킹에서는 총 점수의 랭킹을 보여주고있다. 이지모드와 하드모드에서 각각의 랭킹을 확인 할 수 있는 기능추가.**

1. **더 다양한 문제 난이도 기능 추가**

**현재 덧셈과 곱셈의 문제만 구현이 되어있는데, 더 다양한 난이도와 다양한 문제들의 기능 추가**

1. **틀린문제 모아보기 버그 해결**

**틀린문제가 없을 시 틀린문제모아보기 버튼을 누르면 앱이 종료가 되는 버그를 예외처리를 통해 해결**