

2022학년도 모바일프로그래밍 프로젝트 완료 보고서

# 와이밀리 (whymile)

2022 년 6 월 15 일

컴퓨터정보과

팀명	새육강
팀장	김연욱(201844045)
팀원	김세용(201844044)



인하공업전문대학  
INHA TECHNICAL COLLEGE  
仁荷工業專門大學

2022년도  
컴퓨터정보과 모바일프로그래밍 프로젝트  
수행 결과보고서

(분야 : Android 기반의 앱 개발)

연구과제명 : Android 기반의 챗봇(ChatBot) 앱 개발

2022. 6. 15



인하공업전문대학  
INHA TECHNICAL COLLEGE  
仁荷工業專門大學

본 과제(결과물)는 인하공업전문대학 컴퓨터정보과 모바일프로그래밍 교과목의 프로젝트 완료보고서 입니다.

## 제 출 문

- 분 야 : Android 기반의 소프트웨어 개발
- 연구과제명 : 와이밀리(whymile)
- 프로젝트 책임자 : 인하공업전문대학 컴퓨터정보과 김연욱

2022년도 인하공업전문대학 컴퓨터정보과  
모바일프로그래밍 교과목의 프로젝트 결과물로 이 보고서를 제출합니다.

2022. 6. 15

프로젝트 책임자 : 컴퓨터정보과(201844045) 김 연 욱 (인)

공동 참여자 : 컴퓨터정보과(201844044) 김 세 용 (인)

인하공업전문대학 컴퓨터정보과 학과장 귀하

# 프로젝트 요약문

분 야	Android 기반의 소프트웨어 개발
프로젝트명	와이밀리(whymile)
프로젝트 책임자	컴퓨터정보과(201844045) 김연욱

## 연구 내용



[그림 1] 와이밀리 이미지

본 프로젝트에서는 다수의 사용자가 이용할 수 있는 챗봇 시스템을 구축한다. Android Studio에서 안드로이드 앱을 제작하고, Firebase의 실시간 데이터베이스를 활용하여 데이터를 처리한다. 해당 챗봇 시스템은 머신 러닝(Machine Learning)이나 딥 러닝(Deep Learning)과 같은 학습 형태의 시스템이 아닌 개발자가 작성한 코드 알고리즘 내에서 처리되어 결과를 도출하는 형태의 인공지능 시스템이다. 사용자는 회원가입 이후 로그인을 하여 해당 시스템을 이용할 수 있으며, 회원가입 시에는 무분별한 가입 시도를 방지하기 위하여 Google API reCAPTCHA를 이용했다. 로그인에 성공할 경우에는 시스템을 이용할 수 있는 MenuActivity로 이동한다. 와이밀리(whymile)는 사용자의 질문과 답변을 받는다. 이때 질문에 대한 입력은 어절로 나누어 처리하는 형태이며, 답변은 사용자들의 질문에 대한 입력 값이다. 본 시스템은 사용자들의 입력으로 질문과 답변이 생성되는 형태이므로 정확한 정보를 얻을 수도 있지만 허무맹랑한 정보를 얻을 수도 있다. 이러한 재미있는 요소를 고려한 시스템이 바로 와이밀리(whymile)이다.

## 참여자 인적사항

### 1. 프로젝트 팀장

성명	김연욱	학번	201844045
학과	컴퓨터정보과		
연락처	H.P	010-2560-4658	
	E-mail	kimyeonwook511@gmail.com	

### 2. 프로젝트 팀원

성명	김세용	학번	201844044
학과	컴퓨터정보과		
연락처	H.P	010-3648-8120	
	E-mail	seyong0218@naver.com	

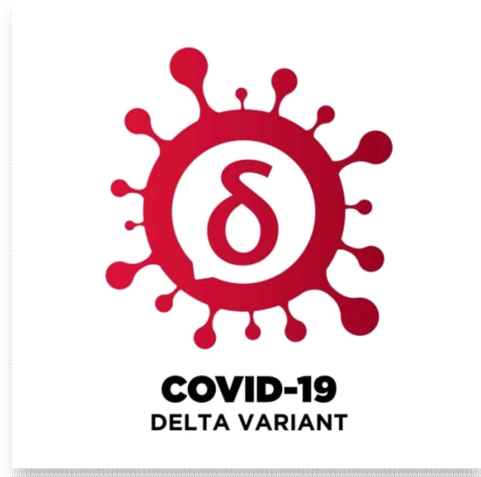
# 목 차

1. 프로젝트 목적	1
2. 프로젝트 목표와 기대 효과	3
3. 프로젝트 진행범위 및 방법	4
4. 프로젝트 주요 내용	5
5. 연구 결과물	8
5-1. 프로젝트명	8
5-2. 프로젝트 개요	8
5-3. 프로젝트 수행 세부 일정 및 내용	8
5-4. 프로젝트 결과	10

## - 부 록 -

프로젝트 소스	10
Firebase	30
사용자 설명서	33

## 1. 프로젝트 목적



[그림 2] 코로나 델타 변이

2019년 중국에서 발견된 코로나 바이러스는 폐렴과 같이 사람들의 폐 기능을 손상시키는 치명적인 바이러스이다. 전염률이 높은 코로나 바이러스는 현재 2022년까지도 이어지고 있다. 이로 인해 많은 사람이 외출을 자제하고 집에서 활동하는 시간이 늘어났다. 이 가운데 사람들은 무료함이나 우울감을 해소하기 위해 여러 가지 방안을 모색하고 있다.



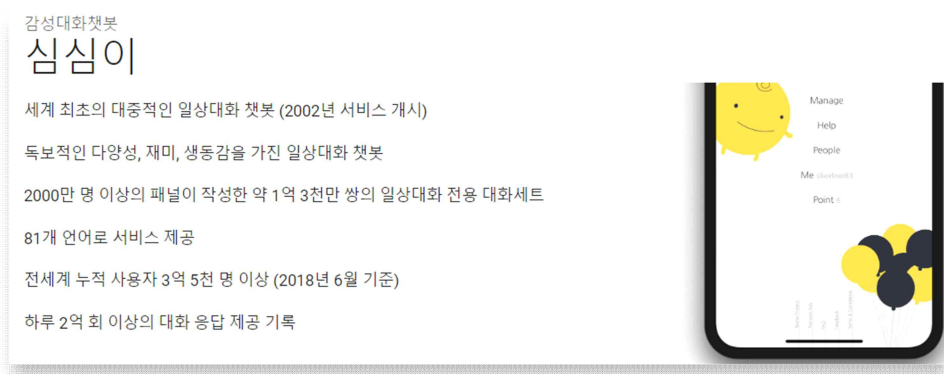
[그림 3] 심심이

여러 가지 방안 중 인공지능과 대화하여 해소하는 방안을 찾는 사람이 늘어나고 있다. 현재 기가지니, 시리, 빅스비, 심심이 등 많은 인공지능 형태의 시스템이 존재한다. 사용자가 시덥지 않은 농담을 하거나, 궁금한 내용에 대한 질문을 하는 등 여러 대화를 인공지능에게 요청하여 질문에 대한 답변을 얻게 된다. [그림 3]에서의 심심이 같은 경우에는 빅데이터를 이용하여 사용자의 질문에 대한 여러 가지 답변을 제공한다.



[그림 4] 신의 탑 웹툰 중 에밀리

[그림 4]는 네이버 웹툰 중 신의 탑에서 나온 인공지능 채팅 로봇이다. 본 웹툰에서는 해당 채팅 로봇인 에밀리를 이용하여 유익한 정보를 얻기도 하고, 시련 속에서 에밀리와의 대화를 통하여 재미를 느끼곤 한다. 그중 에밀리가 제공하는 말도 안되는 정보에 대한 무한한 신뢰를 보이는 웹툰 캐릭터가 있는데, 이 점에서 영감을 얻고 해당 프로젝트를 진행하기로 하였다.



[그림 5] 심심이 사용률

[그림 5]는 심심이의 입력 데이터 개수, 누적 사용자 등이 나와 있다. 많은 사람이 사용하고 즐기는 것을 확인할 수 있다. 심심이를 사용해 보면 질문에 대한 답변이 나온다. 그 답변은 사용자가 직접 학습시키거나 사용자들의 답변 입력을 저장하여 출력하는 형태이며, 이때 입력에 대한 조건을 걸어 학습하는 형태로 데이터를 저장하고 출력한다.



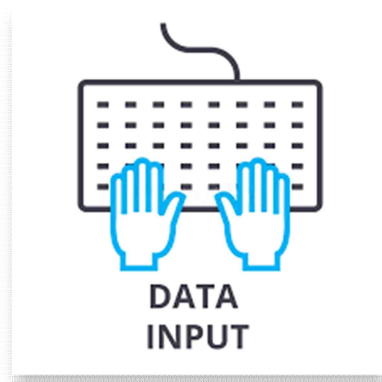
## 2. 프로젝트 목표 및 기대효과

### 2-1 프로젝트 목표

본 프로젝트를 진행하면서 자바 언어에 대한 학습 증진과 Android Studio의 사용법 및 안드로이드 앱의 설계, 실행 과정, UI, UX 등을 학습할 수 있게 된다.

본 프로젝트의 핵심 내용은 머신 러닝(Machine Learning)과 딥 러닝(Deep Learning)처럼 자가적으로 시스템이 학습하는 형태가 아닌 알고리즘을 통한 학습 데이터를 저장하는 형태이다.

### 2-2 프로젝트 기대효과

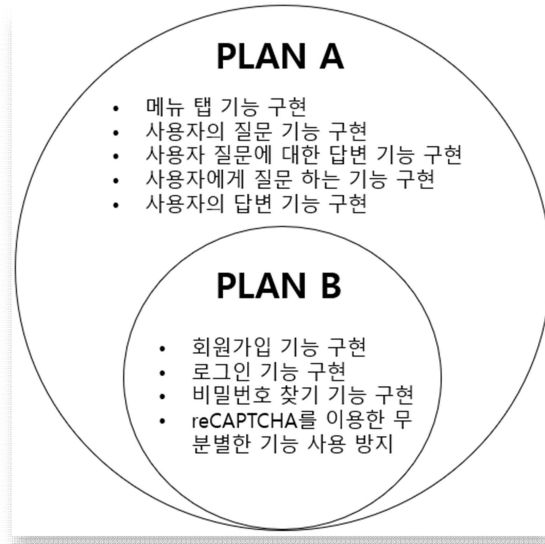


[그림 6] 사용자 입력

본 프로젝트는 사용자들의 입력을 개발자가 설계한 알고리즘에 의해 시스템의 답변이 출력되는 형태이다. 이 챗봇 시스템은 정확한 정보를 주고자 하는 시스템이 아니다. 시스템이 출력하는 답변은 정확한 정보일 수도 있고, 잘못된 정보일 수도 있다. 이는 사용자가 입력하는 데이터에 따라 답변이 달라지기 때문이다. 이 부분을 재미로 생각하여 사용하는 사용자가 있을 수도 있고, 정확한지 정확하지 않은지 모르는 정보에 대해 신뢰하며 사용하는 사용자 또한 있을 수도 있다. 이렇듯 와이밀리라는 챗봇 시스템을 사용하는 이유가 다양할 수 있으므로, 많은 사용자의 정보를 수집할 수 있는 기대효과가 있다.

### 3. 프로젝트 진행범위 및 방법

#### 3-1 프로젝트 진행범위



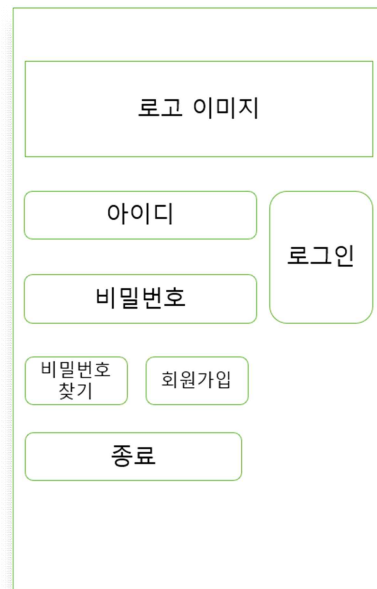
[그림 7] 프로젝트 목표

본 프로젝트는 [그림 7]과 같이 PLAN B와 PLAN A로 나누어 진행한다. PLAN B는 본 프로젝트의 최소 진행 목표로 기본적인 사용자의 로그인 형태를 구현하기 위한 기능들로 나열되어 있다. 사용자는 회원가입 기능을 통해 유저 정보를 생성하고, 해당 정보를 로그인 기능에 사용하여 로그인하는 방식을 제공한다. 추가로 사용자의 비밀번호를 찾기 위한 기능을 구현하고 무분별한 기능 사용을 막기 위한 Google API중 하나인 reCAPTCHA를 사용하도록 한다. PLAN A는 본 프로젝트의 최종 진행 목표로 주요 기능이 포함되어 있다. 우선 챗봇 시스템을 이용하기 위한 로그인 이후 Activity창인 메뉴 탭 기능이 있다. 그리고 챗봇 시스템의 주요 기능 4가지에 해당되는 사용자의 질문, 챗봇의 답변, 챗봇의 질문, 사용자의 답변이 있다. 나열한 순서대로 한 사이클이 진행되며, 사이클 종료 시 대화를 다시 시작하거나 앱을 종료하는 기능을 추가하려 한다.

#### 3-2 프로젝트 진행방법

3-1에서의 [그림 7]에 작성된 목표 기능을 한 개를 구현한 뒤 테스트를 하여 문제점을 찾아가며 프로젝트를 진행해 나간다. 코드에 대한 공유는 메신저로 공유하며, 수정사항이 생길 경우 바로 보고하는 형태로 프로젝트를 진행한다. 프로젝트를 진행하면서 애로사항에 대한 부분은 서로의 의견을 통합하여 해결해 나간다. 마지막으로 참고할 서적 및 교안은 학기 중 공유받은 수업 자료(PDF)파일과 인터넷 검색 등을 통해 정보를 얻기로 한다.

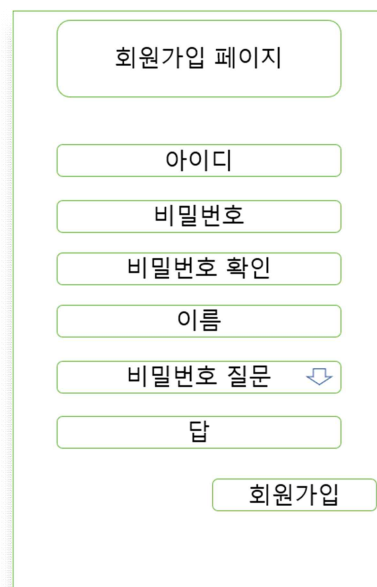
#### 4. 프로젝트 주요 내용



The diagram shows a login screen layout. At the top is a box labeled '로고 이미지' (Logo Image). Below it are two input fields: '아이디' (ID) and '비밀번호' (Password). To the right of these fields is a '로그인' (Login) button. Below the password field is a '비밀번호 찾기' (Find Password) button. To the right of that is a '회원가입' (Sign Up) button. At the bottom is a '종료' (End) button.

[그림 8] 로그인 화면

[그림 8]은 로그인 화면으로 앱 실행시 메인 화면이다. 해당 화면에서는 로그인 기능, 비밀번호 찾기 기능, 회원가입 기능, 종료기능이 있다.



The diagram shows a sign-up screen layout. At the top is a box labeled '회원가입 페이지' (Sign-up Page). Below it are several input fields: '아이디' (ID), '비밀번호' (Password), '비밀번호 확인' (Confirm Password), '이름' (Name), '비밀번호 질문' (Password Question) with a dropdown arrow, and '답' (Answer). At the bottom right is a '회원가입' (Sign-up) button.

[그림 9] 회원가입 화면

[그림 9]는 회원가입 화면으로 입력한 정보에 대한 회원 정보를 생성하는 부분이다. 회원가입 요청 시 아이디 중복, 비밀번호 검사가 이뤄진다.

The image shows a vertical rectangular box representing a web page. At the top, there is a rounded rectangle containing the text "비밀번호 찾기 페이지". Below this, there are four input fields, each a rounded rectangle: "아이디", "비밀번호 질문" (with a small blue dropdown arrow on the right), and "답". At the bottom right of the box, there is a button labeled "비밀번호 찾기".

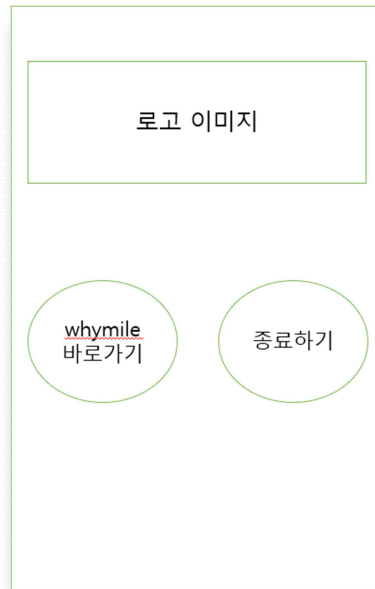
[그림 10] 비밀번호 찾기 화면

[그림 10]은 비밀번호 찾기 화면으로 잊어버린 비밀번호를 찾기 위해 회원가입시 입력한 아이디와 선택한 비밀번호 질문과 이에 대한 답변을 사용하여 비밀번호 찾기를 실행할 수 있다. 여기서 정상적으로 비밀번호 찾기가 이루어 지면 비밀번호 변경 화면으로 이동하게 된다.

The image shows a vertical rectangular box representing a web page. At the top, there is a rounded rectangle containing the text "비밀번호 변경 화면". Below this, there are two input fields, each a rounded rectangle: "변경할 비밀번호" and "비밀번호 확인". At the bottom right of the box, there is a button labeled "비밀번호 변경".

[그림 11] 비밀번호 변경 화면

[그림 11]은 비밀번호 변경 화면으로 변경할 비밀번호와 한번 더 입력한 비밀번호를 확인 하는 입력란이 있다.



[그림 12] 메뉴 선택 화면

[그림 12]는 메뉴 선택 화면으로 성공적으로 로그인하면 보여지는 화면이다. 여기서 whymile 바로가기를 누르면 whymile와 채팅을 할 수 있는 화면으로 이동되며, 종료하기를 누를 시 앱이 정상적으로 종료된다.



[그림 13] whymile 채팅 화면

[그림 13]은 whymile와 채팅하는 화면이다. 사용자가 먼저 whymile에게 질문하며 whymile는 해당 질문에 대한 답변을 한 뒤 반대로 사용자에게 질문을 한다. 사용자는 해당 질문에 대해 답변을 하면 채팅이 종료가 된다.

## 5. 프로젝트 결과물

### 5-1. 프로젝트명

와이밀리(whymile)

(Android 기반의 챗봇 앱 개발)

### 5-2. 프로젝트 개요

본 프로젝트는 로그인을 해야지만 사용할 수 있는 챗봇 시스템이다. 사용자는 회원가입을 통해 사용자 정보를 생성하고 생성된 정보를 사용하여 로그인을 한다. 로그인 후 정상적으로 이뤄지면 챗봇 시스템을 사용하여 원하는 질문과 시스템의 질문에 대한 사용자의 답변을 입력할 수 있다.

### 5-3. 프로젝트 수행 세부일정 및 내용

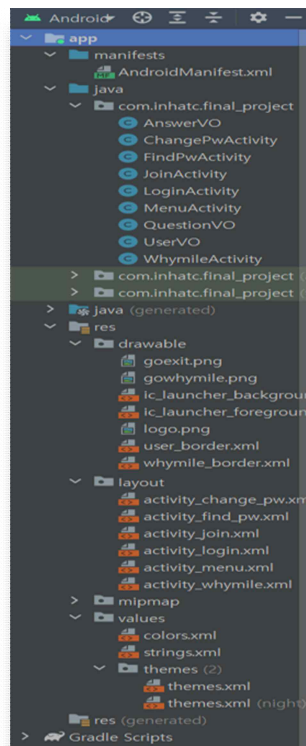
#### ■ 프로젝트 수행 세부 일정

프로젝트 진행 계획(2022. 5. 31. - 2022. 6. 15 )								
진 행 내 용	책임자	1	2	3	4	5	6	비고
벤치마킹 및 설계	김연욱 김세용							
로그인 및 회원가입 Activity 디자인	김연욱 김세용							
로그인 및 회원가입 기능 구현	김연욱 김세용							
비밀번호 찾기 디자인 및 기능 구현, reCAPTCHA연동	김연욱 김세용							
메뉴 탭, 와이밀리 채팅 디자인	김연욱 김세용							
와이밀리 채팅 기능 구현	김연욱 김세용							
시스템 통합 및 테스트	김연욱 김세용							
프로젝트 완료	김연욱 김세용							

## ■ 프로젝트 수행 내용

프로젝트를 진행하며 구현을 목표로 한 기능들은 다음과 같다. (파란 글씨: 구현 성공)

- 회원가입 기능
- 로그인 기능
- 비밀번호 찾기 기능
- 비밀번호 변경 기능
- reCAPTCHA 기능
- 메뉴 선택 기능
- 사용자가 질문하는 기능
- 챗봇이 답변하는 기능
- 챗봇이 질문하는 기능
- 사용자가 답변하는 기능
- 앱 완전 종료하는 기능



[그림 14] 프로젝트 파일

[그림 14]는 프로젝트 파일로 완성된 프로젝트 파일들의 경로를 Android Studio 프로그램 내에서 나타낸 그림이다.

## 5-4. 프로젝트 결과

- 프로젝트 소스 •

### LoginActivity

#### LoginActivity 전역변수

```
1 package com.inhatc.final_project;
2
3 import ...
18
19 public class LoginActivity extends AppCompatActivity implements View.OnClickListener {
20
21     private Button btnLogin, btnFindPw, btnJoin, btnLoginExit;
22     private EditText edtId, edtPw;
23     private DatabaseReference databaseRef;
```

#### LoginActivity 이벤트 & 메소드

```
25 @Override
26 protected void onCreate(Bundle savedInstanceState) {
27     super.onCreate(savedInstanceState);
28     // requestWindowFeature(Window.FEATURE_NO_TITLE); // 타이틀 바 없애기 -> themes.xml에서 설정함
29     setContentView(R.layout.activity_login);
30
31     btnLogin = (Button)findViewById(R.id.btnLogin);
32     btnFindPw = (Button)findViewById(R.id.btnFindPw);
33     btnJoin = (Button)findViewById(R.id.btnJoin);
34     btnLoginExit = (Button)findViewById(R.id.btnLoginExit);
35     edtId = (EditText)findViewById(R.id.edtId);
36     edtPw = (EditText)findViewById(R.id.edtPw);
37
38     btnLogin.setOnClickListener(this);
39     btnFindPw.setOnClickListener(this);
40     btnJoin.setOnClickListener(this);
41     btnLoginExit.setOnClickListener(this);
42
43     databaseRef = FirebaseDatabase.getInstance().getReference(path: "Inhatc_Android_FinalProject");
44 }
```

- Activity 생성 이벤트



```

50      @Override
51      public void onClick(View v) {
52          if (v == btnLogin) {
53              if (edtId.getText().toString().trim().equals("")) {
54                  Toast.makeText(context: this, text: "아이디를 입력해 주세요.", Toast.LENGTH_SHORT).show();
55                  edtId.requestFocus();
56              } else if (edtPw.getText().toString().trim().equals("")) {
57                  Toast.makeText(context: this, text: "비밀번호를 입력해 주세요.", Toast.LENGTH_SHORT).show();
58                  edtPw.requestFocus();
59              } else {
60                  // 로그인 처리
61                  login_user();
62              }
63          }
64      }
65  }
66
67  } else if (v == btnJoin) {
68      // 회원가입 창으로 이동
69      startActivityForResult(new Intent(context: LoginActivity.this, JoinActivity.class), requestCode: 1);
70  }
71  } else if (v == btnFindPw) {
72      // 비밀번호 찾기 창으로 이동
73      startActivityForResult(new Intent(context: LoginActivity.this, FindPwActivity.class), requestCode: 1);
74  }
75  } else if (v == btnLoginExit) {
76      // 앱 종료
77      finishAffinity(); // 루트 액티비티 종료
78      System.runFinalization(); // 작업중인 쓰레드가 다 종료시 종료
79      System.exit(status: 0); // 현재 액티비티 종료
80  }
81  }
82  }

```

- 클릭 이벤트

```

84      @Override
85      protected void onActivityResult(int requestCode, int resultCode, Intent data) {
86          super.onActivityResult(requestCode, resultCode, data);
87          if (data == null) {
88              // 사용자가 뒤로가기 눌렀을 때를 위한 방지
89          }
90          } else if (requestCode == 1) {
91              // 결과 메시지 출력
92              Toast.makeText(context: this, data.getStringExtra(name: "msg"), Toast.LENGTH_SHORT).show();
93          }
94      }
95  }

```

- Activity 이동 후 결과 반환 시 이벤트

```

96 private void login_user() {
97     databaseRef.child("User").addListenerForSingleValueEvent(new ValueEventListener() {
98         @Override
99         public void onDataChange(@NonNull DataSnapshot snapshot) {
100             if (snapshot.child(editId.getText().toString()).child("userId").getValue() == null) {
101                 // 입력된 아이디가 데이터 베이스에 존재하지 않을 경우
102                 Toast.makeText(context: LoginActivity.this, text: "아이디가 존재하지 않습니다.", Toast.LENGTH_SHORT).show();
103             } else if (!snapshot.child(editId.getText().toString()).child("userPw").getValue().toString().equals(editPw.getText().toString())) {
104                 // 입력된 비밀번호가 데이터 베이스에 저장되어 있는 아이디의 비밀번호와 일치하지 않을 경우
105                 Toast.makeText(context: LoginActivity.this, text: "비밀번호가 일치하지 않습니다.", Toast.LENGTH_SHORT).show();
106             } else {
107                 // 로그인 성공
108                 startActivity(new Intent(context: LoginActivity.this, MenuActivity.class));
109             }
110         }
111     });
112 }
113
114 @Override
115 public void onCancelled(@NonNull DatabaseError error) {
116     Toast.makeText(context: LoginActivity.this, text: "로그인 중 에러가 발생했습니다.\nerror : " + error, Toast.LENGTH_SHORT).show();
117 }
118 });
119 }
120 }

```

- 로그인 처리 메소드

- 로그인 처리 중 에러 발생 메소드

## JoinActivity

### JoinActivity 전역변수

```
1 package com.inhatc.final_project;
2
3 import ...
29
30 public class JoinActivity extends AppCompatActivity implements View.OnClickListener {
31
32     private Spinner spinner;
33     private Button btnJoinAct;
34     private EditText edtJoinId, edtJoinPw, edtJoinPwChk, edtJoinName, edtJoinAnswer;
35     private DatabaseReference databaseRef;
36
```

### JoinActivity 이벤트 & 메소드

```
37 @Override
38 protected void onCreate(Bundle savedInstanceState) {
39     super.onCreate(savedInstanceState);
40     setContentView(R.layout.activity_join);
41
42     btnJoinAct = (Button)findViewById(R.id.btnJoinAct);
43     edtJoinId = (EditText)findViewById(R.id.edtJoinId);
44     edtJoinPw = (EditText)findViewById(R.id.edtJoinPw);
45     edtJoinPwChk = (EditText)findViewById(R.id.edtJoinPwChk);
46     edtJoinName = (EditText)findViewById(R.id.edtJoinName);
47     edtJoinAnswer = (EditText)findViewById(R.id.edtJoinAnswer);
48     spinner = (Spinner)findViewById(R.id.spnJoinQuestion);
49
50     String[] items = new String[]{"비밀번호 찾기 질문을 선택해주세요", "나의 제 부를 1호는?", "나의 초등학교는?", "좋아하는 연예인은?"};
51     ArrayAdapter<String> adapter = new ArrayAdapter<>(context: this, android.R.layout.simple_spinner_item, items);
52     adapter.setDropDownViewResource(android.R.layout.simple_spinner_item);
53     spinner.setAdapter(adapter);
54
55     btnJoinAct.setOnClickListener(this);
56
57     databaseRef = FirebaseDatabase.getInstance().getReference(path: "Inhatc_Android_FinalProject");
58
59 }
60
```

- Activity 생성 이벤트

```

61      @Override
62      public void onClick(View v) {
63          if (v == btnJoinAct) {
64              if (edtJoinId.getText().toString().trim().equals("")) {
65                  Toast.makeText(context: this, text: "아이디를 입력해 주세요.", Toast.LENGTH_SHORT).show();
66                  edtJoinId.requestFocus();
67
68              } else if (edtJoinPw.getText().toString().trim().equals("")) {
69                  Toast.makeText(context: this, text: "비밀번호를 입력해 주세요.", Toast.LENGTH_SHORT).show();
70                  edtJoinPw.requestFocus();
71
72              } else if (edtJoinPwChk.getText().toString().trim().equals("")) {
73                  Toast.makeText(context: this, text: "비밀번호 확인을 입력해 주세요.", Toast.LENGTH_SHORT).show();
74                  edtJoinPwChk.requestFocus();
75
76              } else if (edtJoinName.getText().toString().trim().equals("")) {
77                  Toast.makeText(context: this, text: "이름을 입력해 주세요.", Toast.LENGTH_SHORT).show();
78                  edtJoinName.requestFocus();
79
80              } else if (spinner.getSelectedItemPosition() == 0) {
81                  Toast.makeText(context: this, text: "질문을 선택해 주세요.", Toast.LENGTH_SHORT).show();
82                  spinner.setFocusable(true);
83                  spinner.setFocusableInTouchMode(true);
84                  spinner.requestFocus();
85
86              } else if (edtJoinAnswer.getText().toString().trim().equals("")) {
87                  Toast.makeText(context: this, text: "답변을 입력해 주세요.", Toast.LENGTH_SHORT).show();
88                  edtJoinAnswer.requestFocus();
89
90              } else if (!edtJoinPw.getText().toString().trim().equals(edtJoinPwChk.getText().toString().trim())) {
91                  Toast.makeText(context: this, text: "비밀번호가 일치하지 않습니다.", Toast.LENGTH_SHORT).show();
92                  edtJoinPw.setText("");
93                  edtJoinPwChk.setText("");
94                  edtJoinPw.requestFocus();
95
96              } else if (edtJoinId.getText().toString().trim().length() < 6) {
97                  Toast.makeText(context: this, text: "아이디를 6자 이상 입력해 주세요.", Toast.LENGTH_SHORT).show();
98                  edtJoinId.setText("");
99                  edtJoinId.requestFocus();
100
101              } else if (edtJoinPw.getText().toString().trim().length() < 8 || edtJoinPw.getText().toString().trim().length() > 20) {
102                  Toast.makeText(context: this, text: "비밀번호를 8 ~ 20자 입력해 주세요.", Toast.LENGTH_SHORT).show();
103                  edtJoinPw.setText("");
104                  edtJoinPwChk.setText("");
105                  edtJoinPw.requestFocus();
106
107              } else {
108                  UserVO userVO = new UserVO();
109                  userVO.setUserId(edtJoinId.getText().toString().trim());
110                  userVO.setUserPw(edtJoinPw.getText().toString().trim());
111                  userVO.setUserName(edtJoinName.getText().toString().trim());
112                  userVO.setUserQuestion(spinner.getSelectedItem().toString());
113                  userVO.setUserAnswer(edtJoinAnswer.getText().toString().trim());
114
115                  // 매크로 방지 이후 회원가입
116                  reCAPTCHA(userVO);
117
118              }
119          }
120      }

```

- 클릭 이벤트

```

122 @ private void regist_user(UserVO userVO) {
123     databaseRef.child("User").child(userVO.getUserId()).child("userId").addListenerForSingleValueEvent(new ValueEventListener() {
124         @Override
125         public void onDataChange(@NonNull DataSnapshot snapshot) {
126             if (snapshot.getValue() == null) {
127                 databaseRef.child("User").child(userVO.getUserId()).setValue(userVO);
128                 setResult(RESULT_OK, getIntent().putExtra("msg", "회원가입 완료"));
129                 finish();
130             } else {
131                 Toast.makeText(context: JoinActivity.this, text: "이미 아이디가 존재합니다.", Toast.LENGTH_SHORT).show();
132                 edtJoinId.setText("");
133                 edtJoinId.requestFocus();
134             }
135         }
136     }
137 }
138
139
140 @Override
141 public void onCancelled(@NonNull DatabaseError error) {
142     Toast.makeText(context: JoinActivity.this, text: "회원가입 중 에러가 발생했습니다.\nerror : " + error, Toast.LENGTH_SHORT).show();
143 }
144 });
145 }

```

- 회원가입 처리 메소드
- 회원가입 처리 중 에러 발생 메소드

```

147 private void reCAPTCHA(UserVO userVO) { // reCAPTCHA 사용
148     SafetyNet.getClient(activity: JoinActivity.this).verifyWithRecaptcha(siteKey: "6LddimEgAAAAAIr6h8ostCoZaqBzCFieAl_bPCW8")
149         .addOnSuccessListener(new OnSuccessListener<SafetyNetApi.RecaptchaTokenResponse>() {
150             @Override
151             public void onSuccess(SafetyNetApi.RecaptchaTokenResponse recaptchaTokenResponse) {
152                 String captchaToken = recaptchaTokenResponse.getTokenResult();
153
154                 if (captchaToken != null) {
155                     if (!captchaToken.isEmpty()) {
156                         // 보안문자 입력 성공
157                         regist_user(userVO);
158                     } else {
159                         Toast.makeText(context: JoinActivity.this, text: "보안문자 입력에 실패하셨습니다.", Toast.LENGTH_SHORT).show();
160                     }
161                 }
162             }
163         })
164         .addOnFailureListener(new OnFailureListener() {
165             @Override
166             public void onFailure(@NonNull Exception e) {
167                 Toast.makeText(context: JoinActivity.this, text: "Captcha 연결에 실패하셨습니다.", Toast.LENGTH_SHORT).show();
168             }
169         });
170 }
171 }
172 }
173 }

```

- reCAPTCHA 메소드

## JoinActivity에서 사용하는 UserVO

```
1 package com.inhatc.final_project;
2
3 public class UserVO {
4
5     private String userId;
6     private String userPw;
7     private String userName;
8     private String userQuestion;
9     private String userAnswer;
10
11     public String getUserId() { return userId; }
12
13
14     public void setUserId(String userId) { this.userId = userId; }
15
16
17
18     public String getUserPw() { return userPw; }
19
20
21
22     public void setUserPw(String userPw) { this.userPw = userPw; }
23
24
25
26     public String getUserName() { return userName; }
27
28
29
30     public void setUserName(String userName) { this.userName = userName; }
31
32
33
34     public String getUserQuestion() { return userQuestion; }
35
36
37
38     public void setUserQuestion(String userQuestion) { this.userQuestion = userQuestion; }
39
40
41
42     public String getUserAnswer() { return userAnswer; }
43
44
45
46     public void setUserAnswer(String userAnswer) { this.userAnswer = userAnswer; }
47
48
49
50 }
```



## FindPwActivity

### FindPwActivity 전역변수

```
1 package com.inhatec.final_project;
2
3 import ...
24
25 public class FindPwActivity extends AppCompatActivity implements View.OnClickListener {
26
27     private Spinner spinner;
28     private Button btnFindPwAct;
29     private EditText edtFindPwId, edtFindPwAnswer;
30     private DatabaseReference databaseRef;
31
```

### FindPwActivity 이벤트 & 메소드

```
32 @Override
33 protected void onCreate(Bundle savedInstanceState) {
34     super.onCreate(savedInstanceState);
35     setContentView(R.layout.activity_find_pw);
36
37     btnFindPwAct = (Button)findViewById(R.id.btnFindPwAct);
38     edtFindPwId = (EditText)findViewById(R.id.edtFindPwId);
39     edtFindPwAnswer = (EditText)findViewById(R.id.edtFindPwAnswer);
40     spinner = (Spinner)findViewById(R.id.spnFindPwQuestion);
41
42     String[] items = new String[]{"비밀번호 찾기 질문을 선택해주세요", "나의 재 보물 1호는?", "나의 초등학교는?", "좋아하는 연예인은?"};
43     ArrayAdapter<String> adapter = new ArrayAdapter<String>(context: this, android.R.layout.simple_spinner_item, items);
44     adapter.setDropDownViewResource(android.R.layout.simple_spinner_item);
45     spinner.setAdapter(adapter);
46
47     btnFindPwAct.setOnClickListener(this);
48
49     databaseRef = FirebaseDatabase.getInstance().getReference(path: "Inhatec_Android_FinalProject");
50 }
51
```

– Activity 생성 이벤트

```

52      @Override
53      public void onClick(View v) {
54          if (v == btnFindPwAct) {
55              if (edtFindPwId.getText().toString().trim().equals("")) {
56                  Toast.makeText( context: this, text: "아이디를 입력해 주세요.", Toast.LENGTH_SHORT).show();
57                  edtFindPwId.requestFocus();
58              } else if (spinner.getSelectedItemId() == 0) {
59                  Toast.makeText( context: this, text: "질문을 선택해 주세요.", Toast.LENGTH_SHORT).show();
60                  spinner.setFocusable(true);
61                  spinner.setFocusableInTouchMode(true);
62                  spinner.requestFocus();
63              } else if (edtFindPwAnswer.getText().toString().trim().equals("")) {
64                  Toast.makeText( context: this, text: "답변을 입력해 주세요.", Toast.LENGTH_SHORT).show();
65                  edtFindPwAnswer.requestFocus();
66              } else {
67                  // 비밀번호 찾기 처리
68                  find_pw();
69              }
70          }
71      }
72  }
73  }
74  }
75  }

```

- 클릭 이벤트

```

76      private void find_pw() {
77          databaseRef.child("User").child(edtFindPwId.getText().toString()).addListenerForSingleValueEvent(new ValueEventListener() {
78              @Override
79              public void onDataChange(@NonNull DataSnapshot snapshot) {
80                  if (snapshot.getValue() == null) {
81                      // 입력한 아이디가 존재하지 않을 경우
82                      Toast.makeText( context: FindPwActivity.this, text: "입력된 정보가 일치하지 않습니다.", Toast.LENGTH_SHORT).show();
83                      edtFindPwId.setText("");
84                      spinner.setSelection(0);
85                      edtFindPwAnswer.setText("");
86                  } else if (snapshot.child("userQuestion").getValue().toString().equals(spinner.getSelectedItem().toString())
87                      && snapshot.child("userAnswer").getValue().toString().equals(edtFindPwAnswer.getText().toString())) {
88                      // 매크로 방지 후
89                      // 입력한 아이디에 대한 질문과 답변이 일치하는 경우 비밀번호 변경
90                      reCAPTCHA();
91                  } else {
92                      // 입력한 아이디에 대한 질문이나 답변이 일치하지 않을 경우
93                      Toast.makeText( context: FindPwActivity.this, text: "입력된 정보가 일치하지 않습니다.", Toast.LENGTH_SHORT).show();
94                      edtFindPwId.setText("");
95                      spinner.setSelection(0);
96                      edtFindPwAnswer.setText("");
97                  }
98              }
99          });
100      }
101  }
102  }

```

```

103      @Override
104      public void onCancelled(@NonNull DatabaseError error) {
105          Toast.makeText( context: FindPwActivity.this, text: "비밀번호 찾기 중 에러가 발생했습니다.\nerror : " + error, Toast.LENGTH_SHORT).show();
106      }
107  });
108  }
109  }

```

- 비밀번호 찾기 처리 메소드

- 비밀번호 찾기 처리 중 에러 발생 메소드



```

118     @Override
119     protected void onActivityResult(int requestCode, int resultCode, Intent data) {
120         super.onActivityResult(requestCode, resultCode, data);
121
122         if (requestCode == 1) {
123             // 성공적으로 비밀번호 변경 완료시
124             change_pw(data.getStringExtra( name: "changePw"));
125             setResult(RESULT_OK, getIntent().putExtra( name: "msg", value: "비밀번호 변경 완료"));
126             finish();
127         }
128     }

```

- Activity 이동 후 결과 반환 시 이벤트

```

122     private void change_pw(String changePw) {
123         databaseRef.child("User").child(edtFindPwId.getText().toString()).child("userPw").setValue(changePw);
124     }
125

```

- Activity 이동 후 결과 반환 시 이벤트

```

126     private void reCAPTCHA() { // reCAPTCHA 사용
127         SafetyNet.getClient( activity: FindPwActivity.this).verifyWithRecaptcha( siteKey: "6LddimEgAAAAAIr6h8ostCoZaqBzCFieAL_bPCW8")
128         .addOnSuccessListener(new OnSuccessListener<SafetyNetApi.RecaptchaTokenResponse>() {
129             @Override
130             public void onSuccess(SafetyNetApi.RecaptchaTokenResponse recaptchaTokenResponse) {
131                 String captchaToken = recaptchaTokenResponse.getTokenResult();
132
133                 if (captchaToken != null) {
134                     if (!captchaToken.isEmpty()) {
135                         // 보안문자 입력 성공
136                         startActivityForResult(new Intent( packageContext: FindPwActivity.this, ChangePwActivity.class), requestCode: 1);
137                     } else {
138                         Toast.makeText( context: FindPwActivity.this, text: "보안문자 입력에 실패하셨습니다.", Toast.LENGTH_SHORT).show();
139                     }
140                 }
141             }
142         })
143         .addOnFailureListener(new OnFailureListener() {
144             @Override
145             public void onFailure(@NonNull Exception e) {
146                 Toast.makeText( context: FindPwActivity.this, text: "Captcha 연결에 실패하셨습니다.", Toast.LENGTH_SHORT).show();
147             }
148         });
149     }
150 }
151
152

```

- reCAPTCHA 메소드

## ChangePwActivity

### ChangePwActivity 전역변수

```
1 package com.inhatc.final_project;
2
3 import ...
4
16
17 public class ChangePwActivity extends AppCompatActivity implements View.OnClickListener {
18
19     private Button btnChangePwAct;
20     private EditText edtChangePw, edtChangePwChk;
21     private DatabaseReference databaseRef;
```

### ChangePwActivity 이벤트 & 메소드

```
23 @Override
24 protected void onCreate(Bundle savedInstanceState) {
25     super.onCreate(savedInstanceState);
26     setContentView(R.layout.activity_change_pw);
27
28     btnChangePwAct = (Button)findViewById(R.id.btnChangePwAct);
29     edtChangePw = (EditText)findViewById(R.id.edtChangePw);
30     edtChangePwChk = (EditText)findViewById(R.id.edtChangePwChk);
31
32     btnChangePwAct.setOnClickListener(this);
33
34     databaseRef = FirebaseDatabase.getInstance().getReference( path: "Inhatc_Android_FinalProject");
35 }
36
```

#### - Activity 생성 이벤트

```
37 @Override
38 public void onClick(View v) {
39     if (v == btnChangePwAct) {
40         if (edtChangePw.getText().toString().trim().equals("")) {
41             Toast.makeText( context: this, text: "변경할 비밀번호를 입력해 주세요.", Toast.LENGTH_SHORT).show();
42             edtChangePw.requestFocus();
43         } else if (edtChangePwChk.getText().toString().trim().equals("")) {
44             Toast.makeText( context: this, text: "비밀번호 확인을 입력해 주세요.", Toast.LENGTH_SHORT).show();
45             edtChangePwChk.requestFocus();
46         } else if (!edtChangePw.getText().toString().trim().equals(edtChangePwChk.getText().toString().trim())) {
47             Toast.makeText( context: this, text: "비밀번호가 일치하지 않습니다.", Toast.LENGTH_SHORT).show();
48             edtChangePw.setText("");
49             edtChangePwChk.setText("");
50             edtChangePw.requestFocus();
51         } else if (edtChangePw.getText().toString().trim().length() < 8 || edtChangePw.getText().toString().trim().length() > 20) {
52             Toast.makeText( context: this, text: "변경할 비밀번호를 8 ~ 20자 입력해 주세요.", Toast.LENGTH_SHORT).show();
53             edtChangePw.setText("");
54             edtChangePwChk.setText("");
55             edtChangePw.requestFocus();
56         } else {
57             // 변경할 비밀번호 반환
58             setResult(RESULT_OK, getIntent().putExtra( name: "changePw", edtChangePw.getText().toString()));
59             finish();
60         }
61     }
62 }
63
64 }
65
66 }
67
```

- 클릭 이벤트

## MenuActivity

MenuActivity 전역변수

```
1 package com.inhatc.final_project;
2
3 import ...
4
5
6
7
8
9
10 public class MenuActivity extends AppCompatActivity implements View.OnClickListener {
11
12     private ImageView imgGoWhymile, imgExit;
```

MenuActivity 이벤트 & 메소드

```
14 @Override
15 protected void onCreate(Bundle savedInstanceState) {
16     super.onCreate(savedInstanceState);
17     setContentView(R.layout.activity_menu);
18
19     imgGoWhymile = (ImageView)findViewById(R.id.imgGoWhymile);
20     imgExit = (ImageView)findViewById(R.id.imgExit);
21
22     imgGoWhymile.setOnClickListener(this);
23     imgExit.setOnClickListener(this);
24 }
```

- Activity 생성 이벤트

```
26 @Override
27 public void onClick(View v) {
28     if (v == imgGoWhymile) {
29         startActivity(new Intent( packageContext, MenuActivity.this, WhymileActivity.class));
30
31     } else if (v == imgExit) {
32         // 앱 종료
33         finishAffinity(); // 루트 액티비티 종료
34         System.runFinalization(); // 작업중인 쓰레드가 다 종료시 종료
35         System.exit( status, 0); // 현재 액티비티 종료
36
37     }
38 }
39 }
```

- 클릭 이벤트

## WhymileActivity

### WhymileActivity 전역변수

```
1 package com.inhatc.final_project;
2
3 import ...
4
23
24 public class WhymileActivity extends AppCompatActivity implements View.OnClickListener {
25
26     private Button btnSubmit, btnAgain, btnExit;
27     private EditText edtInput;
28     private TextView txtWhymile2, txtWhymile3, txtWhymile4, txtWhymile5, txtWhymile6, txtUser1, txtUser2;
29     private int inputCnt = 0; // 사용자의 입력 횟수
30     private DatabaseReference databaseRef;
31 }
```

### WhymileActivity 이벤트 & 메소드

```
32 @Override
33 protected void onCreate(Bundle savedInstanceState) {
34     super.onCreate(savedInstanceState);
35     setContentView(R.layout.activity_whymile);
36
37     btnSubmit = (Button) findViewById(R.id.btnSubmit);
38     btnAgain = (Button) findViewById(R.id.btnAgain);
39     btnExit = (Button) findViewById(R.id.btnExit);
40     edtInput = (EditText) findViewById(R.id.edtInput);
41     txtWhymile2 = (TextView) findViewById(R.id.txtWhymile2);
42     txtWhymile3 = (TextView) findViewById(R.id.txtWhymile3);
43     txtWhymile4 = (TextView) findViewById(R.id.txtWhymile4);
44     txtWhymile5 = (TextView) findViewById(R.id.txtWhymile5);
45     txtWhymile6 = (TextView) findViewById(R.id.txtWhymile6);
46     txtUser1 = (TextView) findViewById(R.id.txtUser1);
47     txtUser2 = (TextView) findViewById(R.id.txtUser2);
48
49     // 텍스트 뷰 안보이게 하기
50     txtUser1.setVisibility(View.INVISIBLE);
51     txtUser2.setVisibility(View.INVISIBLE);
52     txtWhymile2.setVisibility(View.INVISIBLE);
53     txtWhymile3.setVisibility(View.INVISIBLE);
54     txtWhymile4.setVisibility(View.INVISIBLE);
55     txtWhymile5.setVisibility(View.INVISIBLE);
56     txtWhymile6.setVisibility(View.INVISIBLE);
57
58     databaseRef = FirebaseDatabase.getInstance().getReference(path: "Inhatc_Android_FinalProject");
59     btnSubmit.setOnClickListener(this);
60     btnAgain.setOnClickListener(this);
61     btnExit.setOnClickListener(this);
62 }
```

- Activity 생성 이벤트

```

64      @Override
65      public void onClick(View v) {
66          if (v == btnSubmit) {
67              if (inputCnt == 0) {
68                  // 첫 번째 메시지 입력 (사용자가 질문하기)
69                  inputAct(editInput.getText().toString().trim()); // 입력에 대한 처리 실행
70
71              } else if (inputCnt == 1) {
72                  // 두 번째 메시지 입력 (질문에대한 답변하기)
73                  inputAct(editInput.getText().toString().trim()); // 입력에 대한 처리 실행
74
75              }
76
77          } else if (v == btnAgain) {
78              startActivity(new Intent( packageContext, WhymileActivity.this, WhymileActivity.class));
79
80          } else if (v == btnExit) {
81              finishAffinity(); // 루트 액티비티 종료
82              System.runFinalization(); // 작업중인 쓰레드가 다 종료시 종료
83              System.exit( status, 0); // 현재 액티비티 종료
84
85          }
86      }
87

```

- 클릭 이벤트

```

88     private void inputAct(String input) {
89         if (inputCnt == 0) {
90             // 사용자의 질문 내용 출력
91             txtUser1.setText(input);
92
93             // 1. 사용자의 질문 어절로 나누어서 관련된 답변 있는지 확인
94             databaseRef.child("Question").addListenerForSingleValueEvent(new ValueEventListener() {
95                 @Override
96                 public void onDataChange(@NonNull DataSnapshot snapshot) {
97                     long dataCnt = snapshot.getChildrenCount(); // PK 번호로 생각
98                     for (int i = 0; i < dataCnt; i++) {
99                         if (snapshot.child(Integer.toString(i + 1)).child("question").getValue().toString()
100                             .equals(input)) {
101                             // 사용자의 질문이 이미 존재할 경우
102
103                             // 질문에 대한 답변 출력
104                             printAnswer(input);
105
106                             // 사용자에게 질문할 내용 출력
107                             printQuestion();
108
109                             return;
110                         }
111                     }
112
113                     // 사용자의 질문이 존재하지 않을 경우
114                     // 질문 저장
115                     QuestionVO questionVO = new QuestionVO();
116                     questionVO.setNo((int) dataCnt + 1);
117                     questionVO.setQuestion(input);
118                     databaseRef.child("Question").child(Integer.toString(questionVO.getNo())).setValue(questionVO);
119
120                     // 질문에 대한 답변 출력
121                     printAnswer(input);
122
123                     // 사용자에게 질문할 내용 출력
124                     printQuestion();
125
126                     return;
127                 }
128             });
129
130             @Override
131             public void onCancelled(@NonNull DatabaseError error) {
132                 Toast.makeText(context, WhymileActivity.this, "사용자의 질문 처리 중 예외가 발생했습니다.\nerror : " + error, Toast.LENGTH_SHORT).show();
133             }
134         }
135     }

```

- 사용자의 입력을 처리하는 메소드 (사용자가 질문을 입력하고 전송한 경우)



```

134     } else if (inputCnt == 1) {
135         // 사용자의 질문 내용 출력
136         txtUser2.setText(input);
137
138         // 1. 질문에 대한 답변 저장
139         databaseRef.child("Answer").addListenerForSingleValueEvent(new ValueEventListener() {
140             @Override
141             public void onDataChange(@NonNull DataSnapshot snapshot) {
142                 AnswerVO answerVO = new AnswerVO();
143                 answerVO.setNo((int) snapshot.getChildrenCount() + 1);
144                 answerVO.setQuestion(txtWhymile4.getText().toString());
145                 answerVO.setAnswer(input);
146                 databaseRef.child("Answer").child(Integer.toString(answerVO.getNo())).setValue(answerVO);
147
148                 visibleTextView(); // 데이터가 다 입력 되면 TextView 보여주기
149             }
150         }
151     }

```

```

152     @Override
153     public void onCancelled(@NonNull DatabaseError error) {
154         Toast.makeText(context, WhymileActivity.this, "사용자의 답변 처리 중 에러가 발생했습니다.\nerror : " + error, Toast.LENGTH_SHORT).show();
155     }
156 }
157
158 }
159

```

- 사용자의 입력을 처리하는 메소드 (사용자가 답변을 입력하고 전송한 경우)

```

160 private void printAnswer(String inputQuestion) { // 질문에 대한 답변을 찾는 메소드
161     databaseRef.child("Answer").addListenerForSingleValueEvent(new ValueEventListener() {
162         @Override
163         public void onDataChange(@NonNull DataSnapshot snapshot) {
164             long dataCnt = snapshot.getChildrenCount();
165             String[] inputWords = inputQuestion.split(regex: " "); // 띄어쓰기 기준으로 스플릿
166             ArrayList<String> answerList = new ArrayList<>(); // 답변을 저장할 리스트
167
168             for (int i = 0; i < dataCnt; i++) {
169                 int containCnt = 0;
170
171                 for (int k = 0; k < inputWords.length; k++) {
172                     if (snapshot.child(Integer.toString(i + 1)).child("question").getValue().toString()
173                         .contains(inputWords[k])) {
174                         // 어절이 질문에 포함되어 있을 시 containCnt 추가
175                         containCnt++;
176                     }
177                 }
178                 // containCnt가 (어절의 수 - 1) 보다 크거나 같을 경우에 해당 질문의 답변 저장하기
179                 if (containCnt >= inputWords.length - 1) {
180                     // 해당 dbQuestion의 답변을 저장
181                     answerList.add(snapshot.child(Integer.toString(i + 1)).child("answer").getValue().toString());
182                 }
183             }
184
185             if (answerList.size() == 0) {
186                 // 해당 질문에 대한 답변이 없는 경우
187                 txtWhymile2.setText("저도 잘 모르겠어요.");
188             } else {
189                 // 해당 질문에 대한 답변이 있는 경우
190                 int randomIndex = (int) (Math.random() * answerList.size()); // 답변 리스트에서 랜덤하게 뽑아낼 인덱스
191
192                 for (int k = 0; k < answerList.size(); k++) {
193                     if (k == randomIndex) {
194                         txtWhymile2.setText(answerList.get(k)); // 랜덤하게 뽑아낸 답변을 출력
195                     }
196                 }
197             }
198         }
199     });
200
201     @Override
202     public void onCancelled(@NonNull DatabaseError error) {
203         Toast.makeText(context: WhymileActivity.this, text: "답변 리스트 생성 중 에러가 발생했습니다.\nerror : " + error, Toast.LENGTH_SHORT).show();
204     }
205 }
206
207

```

- 사용자가 입력한 질문에 대한 답변을 찾는 메소드

1. 사용자의 입력을 어절로 나누어 질문에 대한 답변을 찾는다.  
(띄어쓰기 기준으로 입력을 split하여 Answer테이블에서 각 어절이 포함된 질문이 있는 경우 containCnt를 1씩 늘린다.)
2. containCnt가 총 어절 수에서 -1을 한 값 이상일 경우 해당 질문과 관련된 답변을 출력한다.  
미만일 경우 “저도 잘 모르겠어요.” 출력
3. 해당 질문과 관련된 답변이 다수인 경우 랜덤하게 한 개만 출력



```

208 private void printQuestion() { // 사용자에게 할 질문을 출력하는 메소드
209     databaseRef.child("Question").addListenerForSingleValueEvent(new ValueEventListener() {
210         @Override
211         public void onDataChange(@NonNull DataSnapshot snapshot) {
212             long dataCnt = snapshot.getChildrenCount();
213             ArrayList<String> questionList = new ArrayList<>(); // 질문을 저장할 리스트
214
215             for (int i = 0; i < dataCnt; i++) {
216                 System.out.println(snapshot.child(Integer.toString(i + 1)).child("question").getValue().toString());
217                 questionList.add(snapshot.child(Integer.toString(i + 1)).child("question").getValue().toString());
218             }
219
220             int randomIndex = (int) (Math.random() * questionList.size()); // 질문 리스트에서 랜덤하게 뽑아낼 인덱스
221
222             for (int k = 0; k < questionList.size(); k++) {
223
224                 if (k == randomIndex) {
225                     txtWhymile4.setText(questionList.get(k));
226                 }
227             }
228
229             visibleTextView(); // 데이터가 다 입력 되면 TextView 보여주기
230         }
231     });
232
233     @Override
234     public void onCancelled(@NonNull DatabaseError error) {
235         Toast.makeText(context, WhymileActivity.this, "질문 리스트 생성 중 에러가 발생했습니다.\nerror : " + error, Toast.LENGTH_SHORT).show();
236     }
237 }
238

```

- 사용자에게 할 질문을 찾는 메소드

1. 사용자들이 입력한 질문을 전부 리스트에 추가
2. 리스트 중 랜덤하게 한 개만을 추출하여 출력

```

239 private void visibleTextView() { // TextView를 활성화 시키는 메소드
240     if (inputCnt == 0) {
241         txtUser1.setVisibility(View.VISIBLE); // 유저의 메시지 텍스트 뷰 활성화
242         txtWhymile2.setVisibility(View.VISIBLE); // 와이말리의 메시지 텍스트 뷰 활성화
243         txtWhymile3.setVisibility(View.VISIBLE); // 와이말리의 메시지 텍스트 뷰 활성화
244         txtWhymile4.setVisibility(View.VISIBLE); // 와이말리의 메시지 텍스트 뷰 활성화
245
246     } else if (inputCnt == 1) {
247         txtUser2.setVisibility(View.VISIBLE); // 유저의 메시지 텍스트 뷰 활성화
248         txtWhymile5.setVisibility(View.VISIBLE); // 와이말리의 메시지 텍스트 뷰 활성화
249         txtWhymile6.setVisibility(View.VISIBLE); // 와이말리의 메시지 텍스트 뷰 활성화
250
251         edtInput.setVisibility(View.INVISIBLE); // 사용자 메시지 입력 창 비활성화
252         btnSubmit.setVisibility(View.INVISIBLE); // 전송 버튼 비활성화
253         btnAgain.setVisibility(View.VISIBLE); // 다시하기 버튼 활성화
254         btnExit.setVisibility(View.VISIBLE); // 종료하기 버튼 활성화
255
256         btnAgain.setVisibility(View.VISIBLE); // 다시하기 버튼 활성화
257         // 키보드 비활성화
258         ((InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE)).hideSoftInputFromWindow(edtInput.getWindowToken(), flags);
259         edtInput.setInputType(0); // 입력 방지
260
261     }
262
263     edtInput.setText(""); // 입력 초기화
264     inputCnt++;
265 }
266

```

- 입력 처리 후 컴포넌트들의 활성화 / 비활성화 처리 메소드

## WhymileActivity에서 사용하는 QuestionVO

```
1 package com.inhatc.final_project;
2
3 public class QuestionVO {
4
5     private int no;
6     private String question;
7
8     public int getNo() { return no; }
11
12     public void setNo(int no) { this.no = no; }
15
16     public String getQuestion() { return question; }
19
20     public void setQuestion(String question) { this.question = question; }
23 }
```

## WhymileActivity에서 사용하는 AnswerVO

```
1 package com.inhatc.final_project;
2
3 public class AnswerVO {
4
5     private int no;
6     private String question;
7     private String answer;
8
9     public int getNo() { return no; }
12
13     public void setNo(int no) { this.no = no; }
16
17     public String getQuestion() { return question; }
20
21     public void setQuestion(String question) { this.question = question; }
24
25     public String getAnswer() { return answer; }
28
29     public void setAnswer(String answer) { this.answer = answer; }
32 }
```

## AndroidManifest.xml 설정

```
28     <activity
29         android:name=".LoginActivity"
30         android:exported="true">
31         <intent-filter>
32             <action android:name="android.intent.action.MAIN" />
33
34             <category android:name="android.intent.category.LAUNCHER" />
35         </intent-filter>
36     </activity>
```

- <intent-filter> 생략... </intent-filter>를 LoginActivity에 넣으면서 앱 실행 시 보여지는 Activity를 변경함 (MainActivity -> LoginActivity)

- Firebase

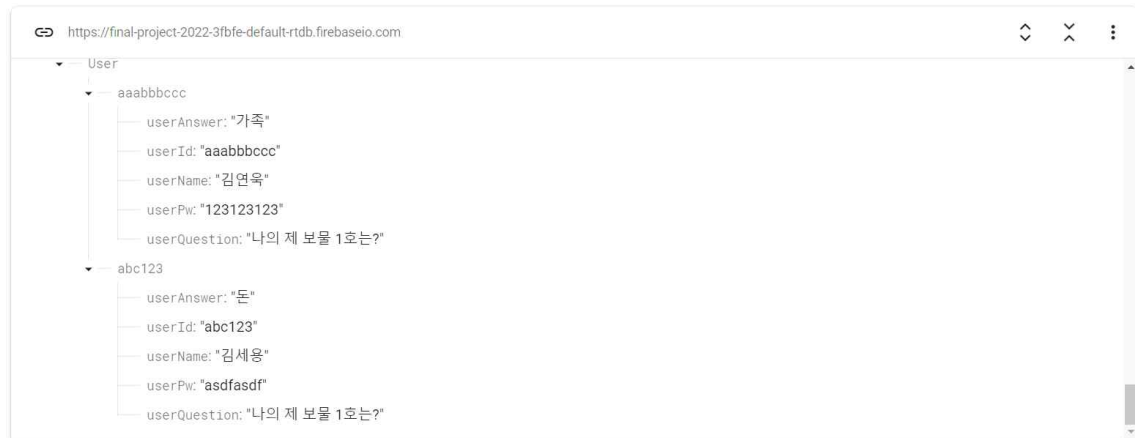
- Answer테이블 (answer : 질문에 대한 답변, no : PK, question : 사용자의 질문)



– Question테이블 (no : PK, question : 사용자의 질문)

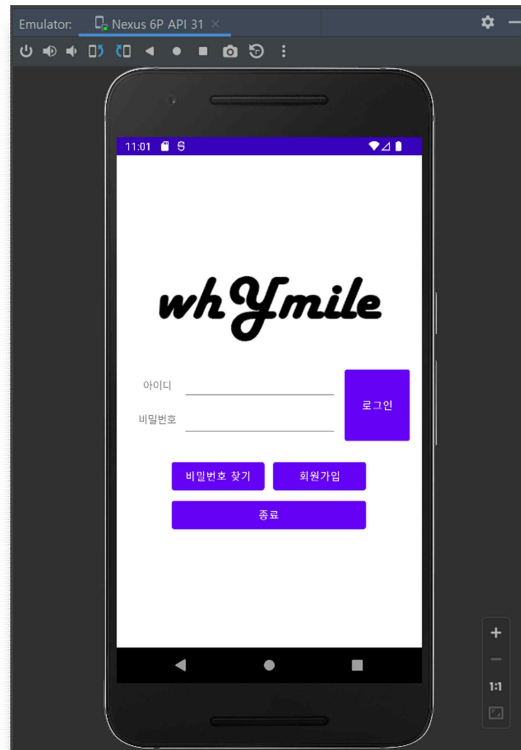
question: "여행 장소 추천해줘"
▼ Question
▼ 1
no: 1
question: "당근 맛있어?"
▼ 2
no: 2
question: "내일 뭐 먹을까?"
▼ 3
no: 3
question: "좋은 노래 알려줘"
▼ 4
no: 4
question: "여행 장소 추천해줘"
▼ 5
no: 5
question: "인후 식당 추천좀"
▼ 6
no: 6
question: "배고픈데 메뉴 추천좀"
▼ 7
no: 7
question: "내일 날씨 어때?"
▼ 8
no: 8
question: "내 생일은 언제게?"
▼ 9
no: 9
question: "오늘 뭐 먹지?"
▼ 10
no: 10
question: "애국가 1절"
▼ 11
no: 11
question: "인후 식당 추천해줘"
▼ 12
no: 12
question: "인후 식당 추천요"
▼ 13
no: 13
question: "애국가 들려줘"
▼ 14
no: 14
question: "오늘 비가 올까?"
▼ 15
no: 15
question: "어디 치킨을 먹을까?"
▼ 16
no: 16
question: "무슨 피자를 먹을까?"
▼ 17
no: 17
question: "인후 식당 추천해줘"
▼ 18
no: 18
question: "인후가 어디야"
▼ 19
no: 19
question: "배고픈데 뭐먹을까"
▼ 20
no: 20
question: "인후 맛집 추천좀 해줘"

- User테이블 (userAnswer : 비밀번호 찾기 답, userId : 아이디, userName : 이름  
userPw : 비밀번호, userQuestion : 비밀번호 찾기 질문)



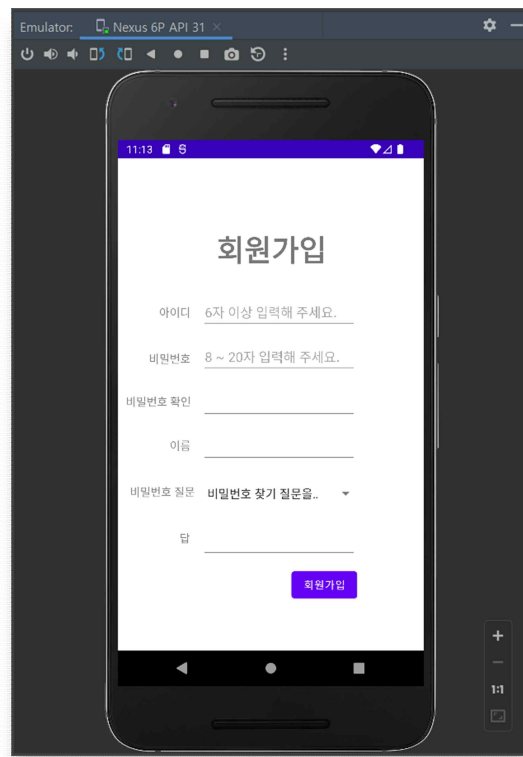
- 사용자 메뉴얼 (부록)

## 1. 로그인 화면



로그인 화면은 앱 실행 시 보이는 메인 화면으로 생성한 아이디와 비밀번호를 통해 로그인을 할 수 있다. 비밀번호를 잊어버렸을 때 비밀번호를 찾는 기능과 아이디를 생성하는 회원가입 기능, 앱을 종료하는 기능이 있다.

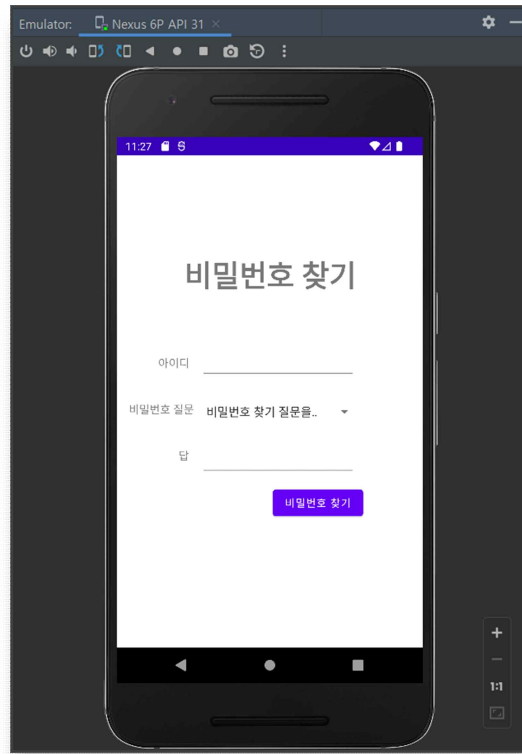
## 2. 회원가입 화면



회원가입 화면은 사용자의 아이디를 성하는 부분이다. 해당 화면에서 입력한 정보로 아이디를 생성할 수 있으며, 비밀번호를 분실했을 시 해당 비밀번호를 찾기 위한 질문과 답변 내용을 입력할 수 있다. 아이디는 6자 이상, 비밀번호는 8~20자를 입력해야 한다. 만약 입력란이 공란일 경우에는 Toast를 이용하여 해당 부분이 입력이 안된 것을 판단하여 알려주고 포커스를 해당 EditText로 이동시킨다.

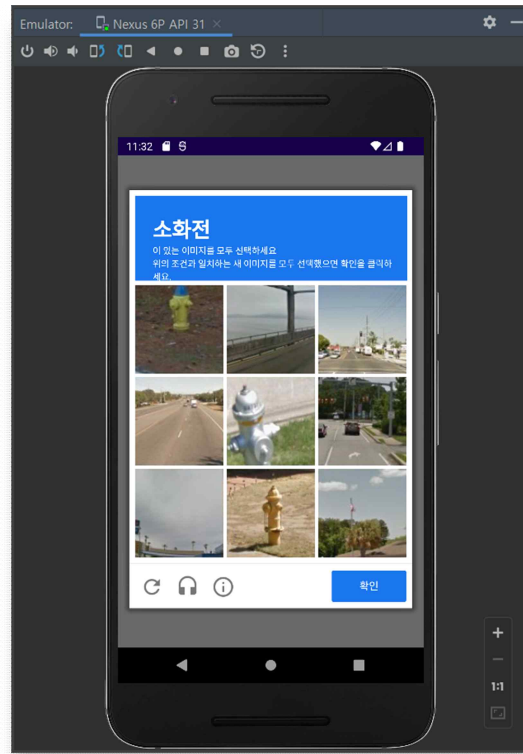


### 3. 비밀번호 찾기 화면



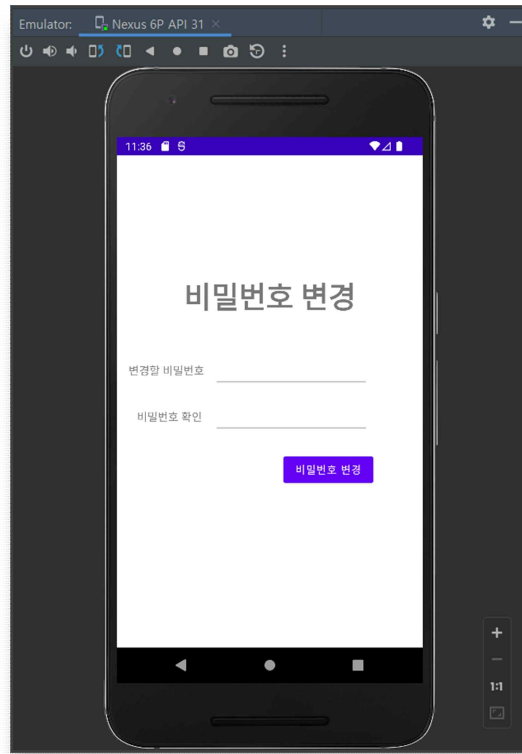
비밀번호 찾기 화면은 가입한 아이디와 선택한 질문, 질문에 대한 답을 통해 비밀번호를 찾을 수 있다. 가입할 때 입력한 비밀번호를 알 수 있는 형태가 아닌 비밀번호를 변경하는 형태로 비밀번호를 찾는다.

#### 4. reCAPTCHA 화면



reCAPTCHA 화면은 사용자의 무분별한 기능 사용을 방지하기 위한 (일명 매크로 방지) 기능으로 Google의 API중 하나인 reCAPTCHA를 사용하였다. 정상적으로 보안문자를 입력하고 통과하면 다음 기능을 수행한다.

## 5. 비밀번호 변경 화면



비밀번호 변경 화면은 비밀번호 찾기 화면에서 정상적으로 입력을 통해 넘어온 화면이다. 해당 부분에서는 변경할 비밀번호를 입력하고 확인하여 아이디에 대한 비밀번호를 변경할 수 있다.

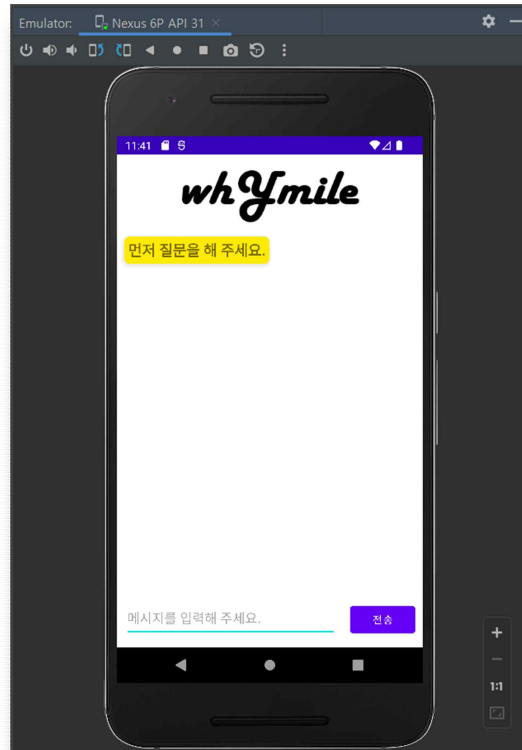
## 6. 메뉴 선택 화면



메뉴 선택 화면에서는 whymile 챗봇을 이용하는 화면으로 이동하는 기능과 앱을 종료하는 기능이 있다.

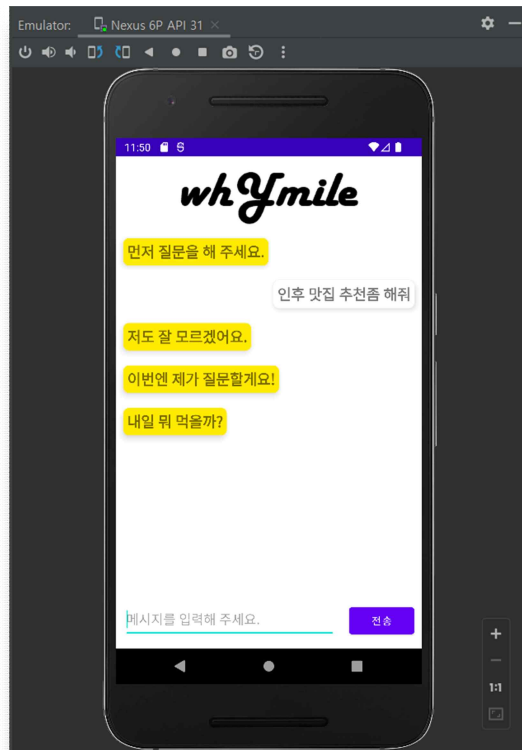
## 7. 와이밀리 채팅 화면

### 7-1 첫 번째 화면



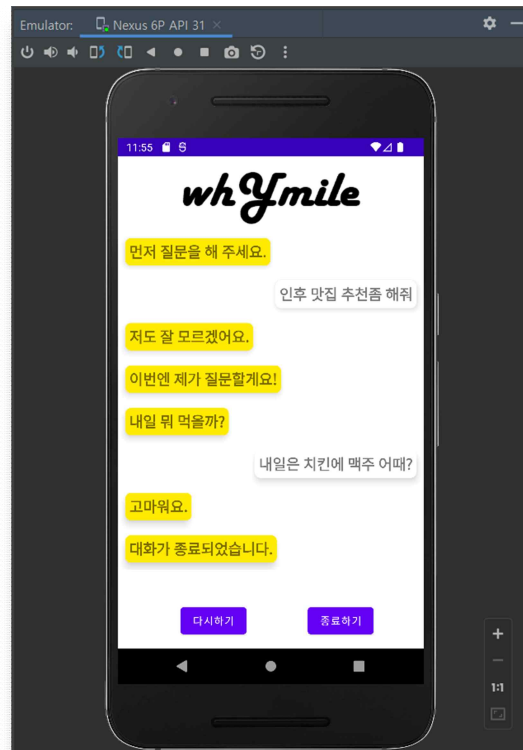
와이밀리 채팅 화면이다. 첫 번째 화면은 위와 같으며 why mile가 “먼저 질문을 해 주세요”라는 메시지를 보낸 채로 시작된다.

## 7-2 두 번째 화면



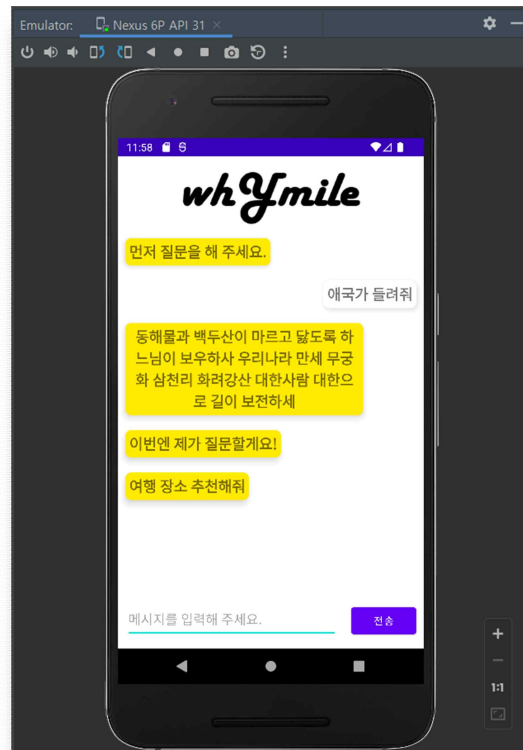
두 번째 화면은 위와 같으며 사용자가 질문을 입력하고 전송한 화면이다. 사용자의 질문에 대해 whymile는 답변을 찾고, 해당 질문에 대해 답변할 내용이 없을 경우 위와 같이 “저도 잘 모르겠어요.”를 출력한다. 그 다음 whymile가 사용자에게 질문을 한다.

### 7-3 세 번째 화면



세 번째 화면은 위와 같으며 whymile의 질문에 대해 사용자가 답변을 한 상태이다. 답변을 완료하였으면 whymile는 “고마워요.”라는 문구를 출력한 뒤 대화를 종료한다. 이때 사용자는 다시 한번 더 대화할 수 있는 다시 하기 버튼과 앱을 종료하는 종료하기 버튼을 사용할 수 있다.

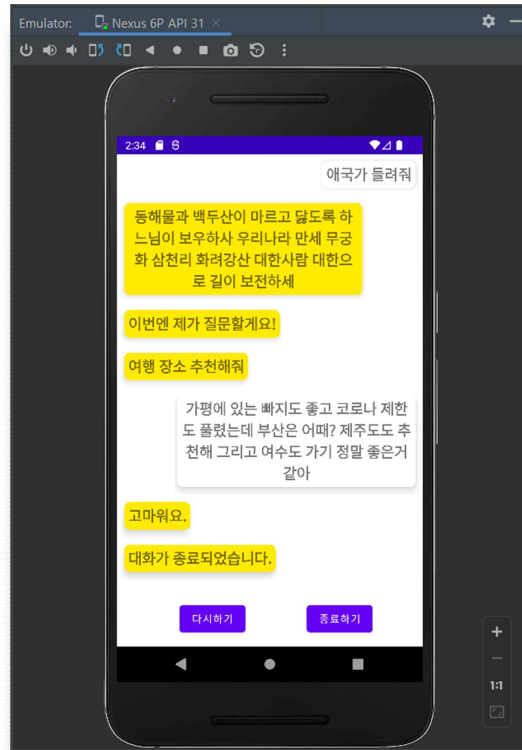
#### 7-4 두 번째 화면에서의 답변이 있는 경우(7-2의 다른 경우)



위 화면은 7-2와는 다르게 질문에 대한 답변이 있는 경우이다. 입력을 어절로 나누어 질문과 관련된 답변을 찾아 출력한다.



## 7-5 스크롤 뷰



위 화면은 출력 데이터가 너무 많을 경우를 생각하여 스크롤 뷰를 사용하여 위아래로 드래그 시 데이터를 이어서 볼 수 있게 구현해 두었다.

## 부 록

1. 프로젝트 Source	10
2. Firebase	30
3. 사용자 설명서	33