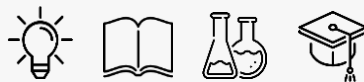


Type Mate

류하준

유승국

이현정



INDEX



개발 동기

애플리케이션 설명
개발 동기



코드 설명

어플 화면
주요 기능 설명



프로젝트 시연

기능 시연

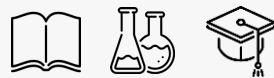


Q&A

질의응답



개발 동기



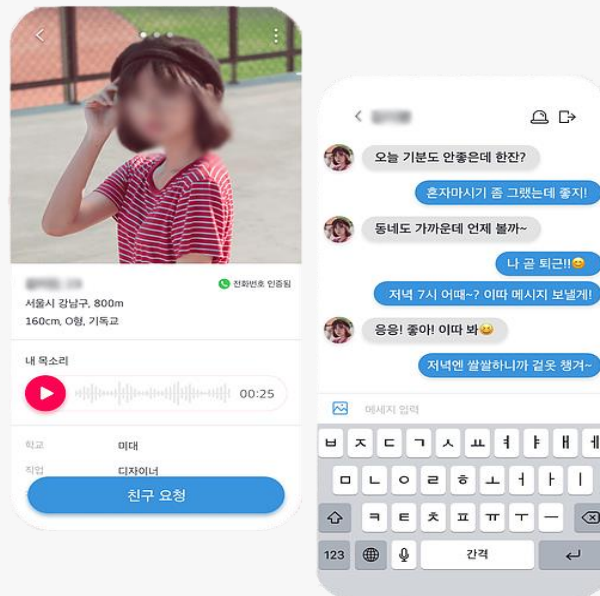


개발 동기

왜 TypeMate라는 어플을 만들었는가?

MBTI

+



기존의 채팅 어플들과 다르게
MBTI로 상대의 성향을 미리 확인하면
대화가 잘 통할 것 같아서 개발하게 되었다.





코드 설명

주요 기능 설명





코드 설명

DB연동의 주요 기능에 대한 코드 설명을 해보자.

DB연동

Oracle

USERS

Column Name	Data Type	Nullable	Default	Primary Key
U_IDX	NUMBER(3,0)	No	-	1
ID	VARCHAR2(50)	No	-	-
PWD	VARCHAR2(50)	No	-	-
NAME	VARCHAR2(50)	No	-	-
NICKNAME	VARCHAR2(50)	Yes	-	-
GENDER	VARCHAR2(20)	No	-	-
BIRTH	VARCHAR2(50)	No	-	-
MBTI	VARCHAR2(10)	No	-	-
TEL	VARCHAR2(50)	No	-	-
1 - 9				

Spring

- controller
 - UserController.java
- dao
 - UserDAO.java
- vo
 - UserVO.java
- config.mybatis.mapper
 - user.xml

Android

- async
 - SpringTask
- util
 - Util
- vo
 - ChatVO
 - UserVO





코드 설명

DB연동의 주요 기능에 대한 코드 설명을 해보자.

DB연동

Android

```
resultStr = new SpringTask(Util.SERVER_IP_LOGIN, context: LoginActivity.this).execute(param).get();  
  
JSONObject jsonObject = new JSONObject(resultStr);
```

Spring

```
@RequestMapping(value = "/login.do", produces = "application/json; charset=UTF8")  
@ResponseBody  
public String login(UserVO vo) {  
  
    . . .  
  
    resultStr = String.format("{\"result\":\"%s\", 'user': {'nickname': '%s', 'gender': '%s', 'birth': %s, 'mbti': '%s'}}",  
                             result, vo.getNickname(), vo.getGender(), vo.getBirth(), vo.getMbti());  
}  
return resultStr;  
}
```





코드 설명

Firebase의 주요 기능에 대한 코드 설명을 해보자.

Firestore – 안드로이드 연동 및 insert

TypeMate ▾

typemate-95306

chat

fbgkwns

user1

-MEWJAgVu9yMK4BnXWXa

message: "feedfg"

nickname: "fbgkwns"

time: "오후 3:17"

-MEWJFIr-Wb5_ksKJka4

message: "hi! :)"

nickname: "user1"

time: "오후 3:17"

user1

fbgkwns

-MEWJAgZDfJct9ysmDP

message: "feedfg"

nickname: "fbgkwns"

time: "오후 3:17"

-MEWJFIqd1vq0D8I55f0

message: "hi! :)"

nickname: "user1"

time: "오후 3:17"

Util

```
static FirebaseDatabase DATABASE = FirebaseDatabase.getInstance(); // 데이터베이스 객체
static DatabaseReference CHAT_REF = DATABASE.getReference(path: "chat"); // chat을 참조하는 데이터베이스 개장객체
```

ChatActivity

```
//firebase\chat\내 닉네임\상대닉네임\ChatVO값
Util.CHAT_REF.child(myNickname).child(opponentNickname).push().setValue(vo); // 서버에 채팅객체 데이터 쓰기
//firebase\chat\상대\내 닉네임\ChatVO값
Util.CHAT_REF.child(opponentNickname).child(myNickname).push().setValue(vo);
```






코드 설명


주요 기능에 대한 코드 설명을 해보자.


회원가입 - 실시간 중복 체크

U+ 3:34 70%

Sign Up

 fbgkwns 사용 불가





 Password 불일치


다음

U+ 3:35 70%

Sign Up

 fbgkwns123 사용 가능



|

다음





코드 설명

주요 기능에 대한 코드 설명을 해보자.

회원가입 - MBTI 검사



```
web_view_join = findViewById(R.id.web_view_join);
web_view_join.loadUrl("http://www.16personalities.com/ko/무료-성격-유형-검사");
// css&javascript 호환문제를 해결해주는 코드
web_view_join.getSettings().setJavaScriptEnabled(true);
web_view_join.setWebViewClient(new WebViewClient() {
    String url = request.getUrl().toString();

    //url에 -e이나 -i가 있을 때 얼럿창 띄우기
    if (url.contains("-e") || url.contains("-i")) {
        // 주소창에 있는 mbti만 가져오기
        type = url.substring(url.indexOf('-') + 1, url.length()).toUpperCase();
        // 확인창 띄우기
        showAlert();
        return true;
    }
    return false;
});
```

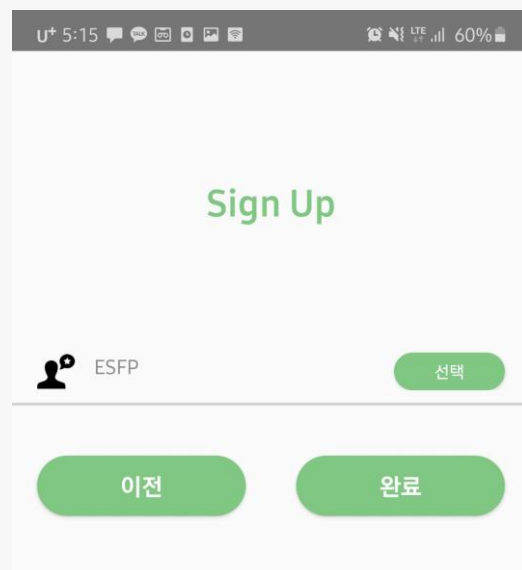
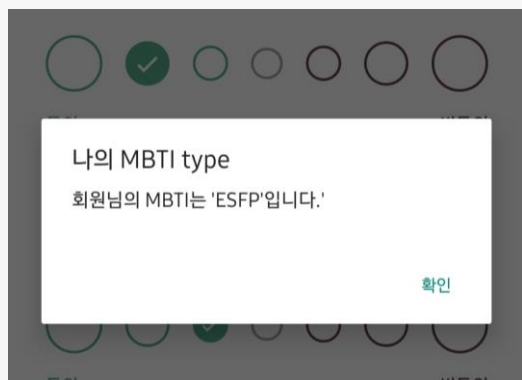




코드 설명

주요 기능에 대한 코드 설명을 해보자.

회원가입 - MBTI 검사



```
private void showAlert() {  
    // 타이틀과 메시지 지정  
    AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder( context: MbtiTestActivity.this);  
    alertDialogBuilder.setTitle("나의 MBTI type").setMessage("회원님의 MBTI는 ' " + type + "'입니다.'");  
  
    // 다이얼로그에 OK 버튼 만들고 감지자 정의  
    alertDialogBuilder.setPositiveButton( text: "확인", (dialogInterface, i) -> {  
        // 다시 JoinActivity로 전환  
        Intent intent = new Intent( packageContext: MbtiTestActivity.this, JoinActivity.class);  
  
        // JoinActivity에 들어갈 항목 다시 보내기  
        intent.putExtra( name: "mbti", value: " " + type);  
  
        . . .  
  
        startActivity(intent);  
        finish();  
    });  
  
    // 설정된 값으로 다이얼로그 최종생성  
    AlertDialog alertDialog = alertDialogBuilder.create();  
    alertDialog.setCancelable(false); // 뒤로가기 버튼, 배경 터치로 인해 꺼짐 방지  
    // 사용자에게 보여줌  
    alertDialog.show();  
}
```





코드 설명

주요 기능에 대한 코드 설명을 해보자.

유형 검색

U+ 5:37 56%

검색할 조건을
선택하세요.

gender

남자 여자

mbti

ISTJ ISFJ

INFJ INTJ

ISTP ISFP

INFP INTP

ESTJ ESFJ

ENFJ ENTJ

ESTP ESFP

ENFP ENTP

검색

son ESTJ

park ESTJ

kim ISTJ

lee ISTJ

user3 ESTJ

user2 ESTJ

Android

```
View.OnClickListener searchClick = (view) -> {  
    if(!btn_man.isChecked() && ! btn_woman.isChecked()) {  
        gender = "";  
    }  
  
    mbti = "";  
  
    //ISFP|INTJ...  
    if (mbtiArr.size() > 0) {  
        for (String arr : mbtiArr) {  
            mbti += arr + "|";  
        }  
        mbti = mbti.substring(0, mbti.length() - 1);  
    }  
  
    s_list.clear();  
    adapter = null;  
    drawer_layout.closeDrawer(drawer);  
  
    String param = "nickname=" + user_nickname + "&gender=" + gender + "&mbti=" + mbti;  
    SearchResult(param);  
}; // searchClick
```

Spring

```
<!-- 검색 -->  
<select id="u_search" parameterType="user" resultType="user">  
    select * from users  
    where nickname != #{nickname} and regexp_like(gender,#{gender}) and regexp_like(mbti,#{mbti}) order by u_idx desc  
</select>
```





코드 설명

주요 기능에 대한 코드 설명을 해보자.

채팅리스트



//chat\내 닉네임의 하위목록에 값 변경에 대한 감지자

```
Util.CHAT_REF.child(myNickname).addChildEventListener(new ChildEventListener() {
```

```
@Override
```

```
public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {
```

```
    chatListVO = new ChatVO();
```

```
//key값으로 존재하는 상대방의 닉네임
```

```
String opponentNickname = snapshot.getKey();
```

```
String url = Util.SERVER_IP_CHECK_GENDER;
```

```
String param = "nickname=" + opponentNickname;
```

```
try {
```

```
//여러개의 PC에서 다른DB 값으로 접근할시에 firebase의 값과 데이터가 혼동되어 null값이 넘어오고 오류가 난다.
```

```
String result = new SpringTask(url, context).execute(param).get();
```

```
chatListVO.setGender(result);
```

```
} catch (Exception e) {
```

```
    Log.i( tag: "CATCH", e.getMessage());
```

```
}
```

```
Map map = (HashMap)snapshot.getValue();
```

```
Iterator<DataSnapshot> iterator = snapshot.getChildren().iterator();
```

```
while (iterator.hasNext()){
```

```
String key = iterator.next().getKey();//HashMap형태를 Iterator로 반환해주는 Children()에서 Key()값을 꺼내온다
```

```
Map result = (HashMap) map.get(key);//HashMap형태로 담겨있는 value값을 map에 담아준 후
```

```
chatListVO.setMessage((String) result.get("message"));//그 안에 key가 message인 value값을 꺼내온다
```

```
chatListVO.setTime((String) result.get("time"));|
```

```
}
```

```
//위에서 추출했던 상대 유저의 닉네임을 VO에 담는다
```

```
chatListVO.setNickname(opponentNickname);
```

```
//상대 유저의 닉네임을 담고있는 VO를 어댑터에 넣어줄 ArrayList에 넣어준다
```

```
chatRoom_arr.add(chatListVO);
```

```
//어댑터에서 값이 변경될때마다 List를 업데이트해줌
```

```
chatRoomAdapter.notifyDataSetChanged();
```

```
}//added()
```



코드 설명

주요 기능에 대한 코드 설명을 해보자.

채팅방

//onCreate()시 디비의 값을 감지해 호출되는 감지자를 만듦

```
Util.CHAT_REF.child(myNickname).child(opponentNickname).addChildEventListener(firebaseEvt);
```

```
ChildEventListener firebaseEvt = new ChildEventListener() {  
    @Override  
    public void onChildAdded(@NonNull DataSnapshot snapshot, @Nullable String previousChildName) {  
        //채팅 보내기  
        ChatVO vo = snapshot.getValue(ChatVO.class);  
        vo.setMsgKey(snapshot.getKey());  
        arr.add(vo);  
        chatAdapter.notifyDataSetChanged();  
        chat_list.setTranscriptMode(ListView.TRANSCRIPT_MODE_ALWAYS_SCROLL);  
        chat_list.setSelection(arr.size());  
    }  
}
```

```
@Override  
public void onChildRemoved(@NonNull DataSnapshot snapshot) {  
    int cnt;  
    cnt = chatAdapter.getCount();  
  
    //어댑터에 아이템이 있을때만 실행  
    if (cnt > 0) {  
        //key값으로 존재하는 상대방의 닉네임이 넘어온다  
        String msgKey = snapshot.getKey();  
  
        //아이템 수정  
        for (int i = 0; i < arr.size(); i++) {  
            if (arr.get(i).getMsgKey().equals(msgKey)) {  
                arr.remove(i);  
                chatAdapter.notifyDataSetChanged();  
            }  
        }  
    }  
}  
} //onChildRemoved()
```

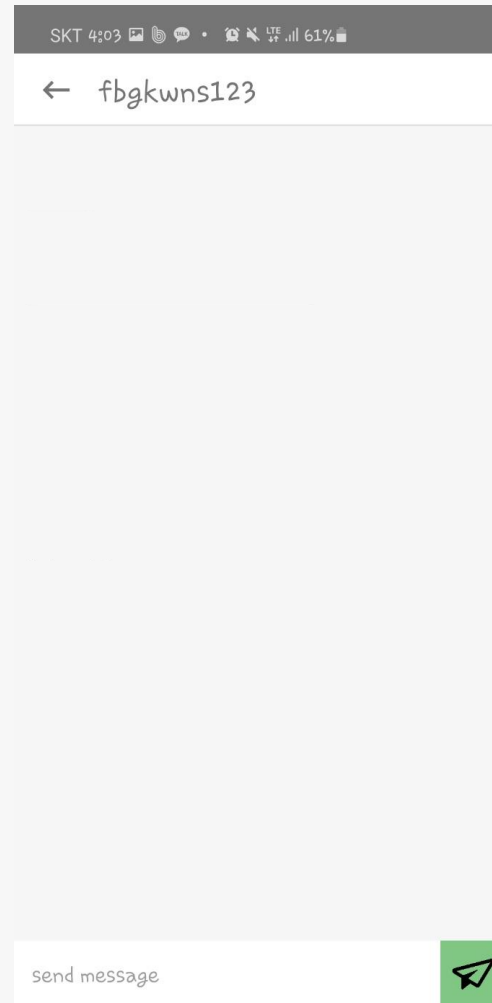
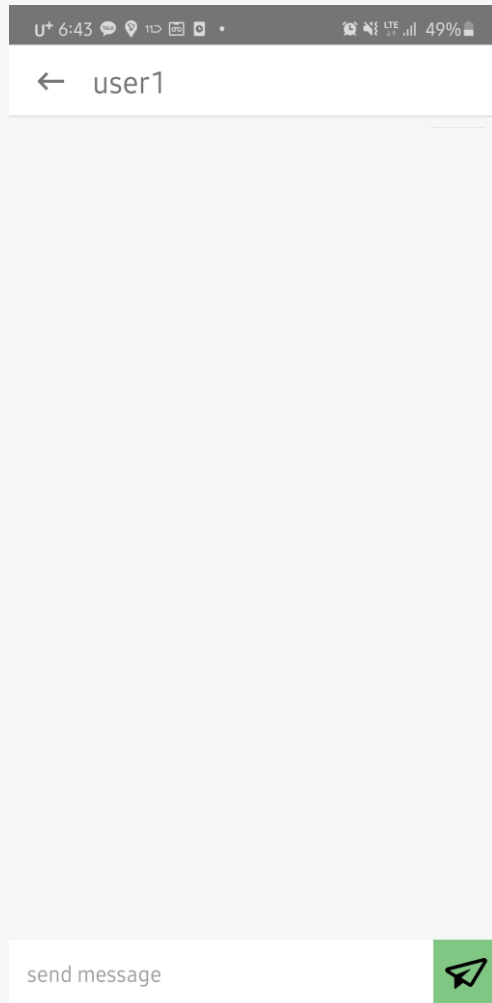




코드 설명

주요 기능에 대한 코드 설명을 해보자.

채팅방





프로젝트 시연

기능 시연

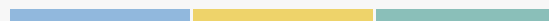




Q & A

질의 응답





THANK YOU

