

20181763 김규식

포트폴리오

졸업작품 포트폴리오

김규식

kuchic0918@naver.com

<https://github.com/kuchic0918>

010-4072-9809





1조 견주개발자

팀장

김규식

팀프로젝트 수행간에 로그인,회원가입,커뮤니티 기능을 구현하였고 기능을 구현하기 위해 서버와 연결하는 작업을 하였습니다.

모두모이개

초보견주들을 위한 알파배기 앱



GRADUATION

2017 경성고등학교 졸업

2018 동양미래대학교 컴퓨터정보공학과 입학

2022 동양미래대학교 컴퓨터정보공학과 졸업

SKILL

JAVA	<div><div></div></div>	95
ANDROID	<div><div></div></div>	80
FIREBASE	<div><div></div></div>	60

개발 Tool

Java

Android Studio

Server

Api

팀원

김시현

오예림

석현우

김송희

목차

C O N T E N T S



01 깃허브

- 협업
- Commit
- push
- project update

02 Firebase

- Firebase 란 ?
- Firebase 연동

03 SNS

- UI/UX
- 기능별 코드

PROJECT.1

GIT HUB



01

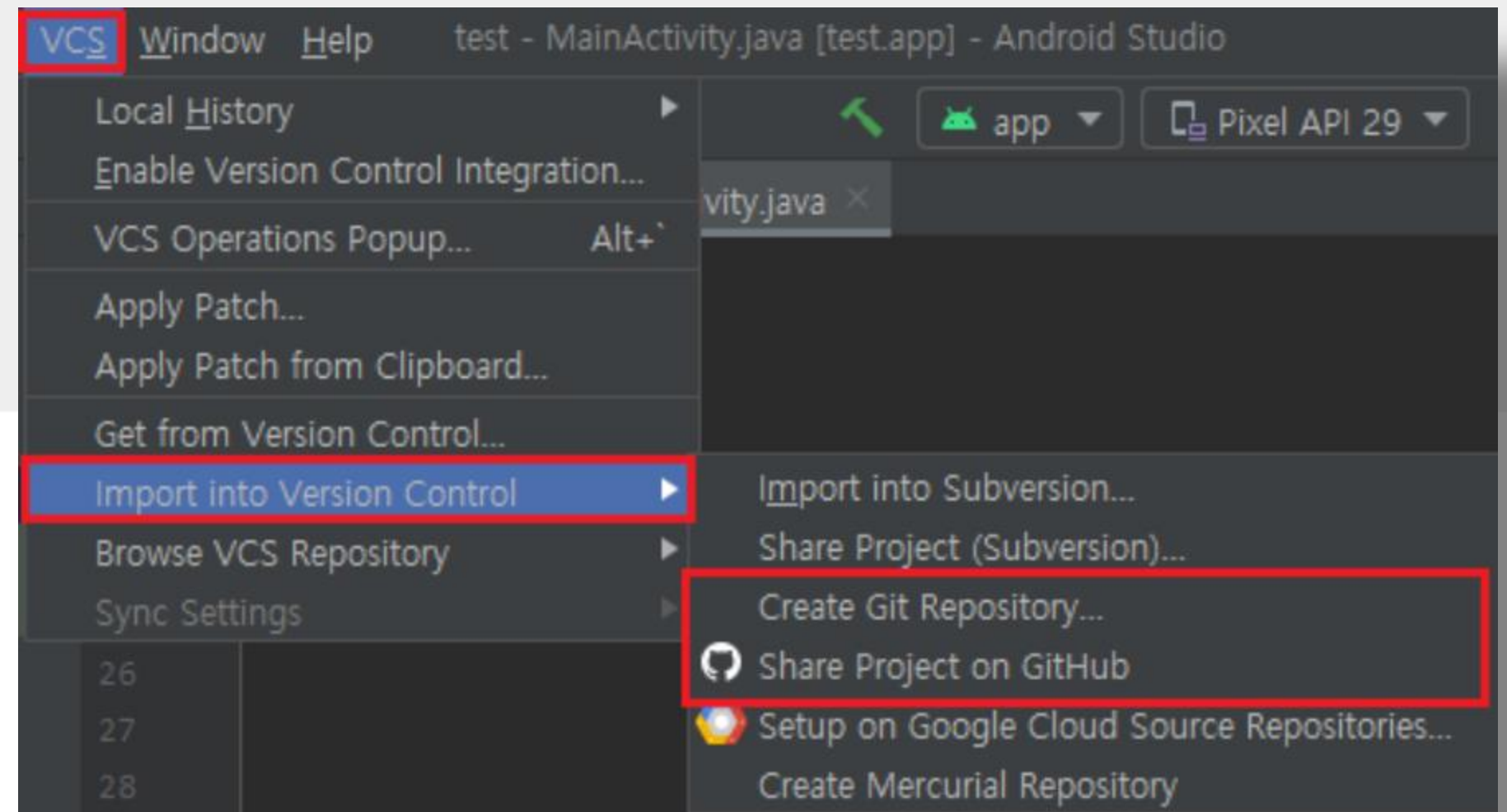
-프로젝트 수정 후 업로드

-협업 과정

Git Hub

프로젝트 업로드

1. Create Git Repository
- 좌측 프로젝트 영역 파일들이 빨강게 바뀐다
2. Share Project on GitHub
3. 깃허브에 업로드가 되었다



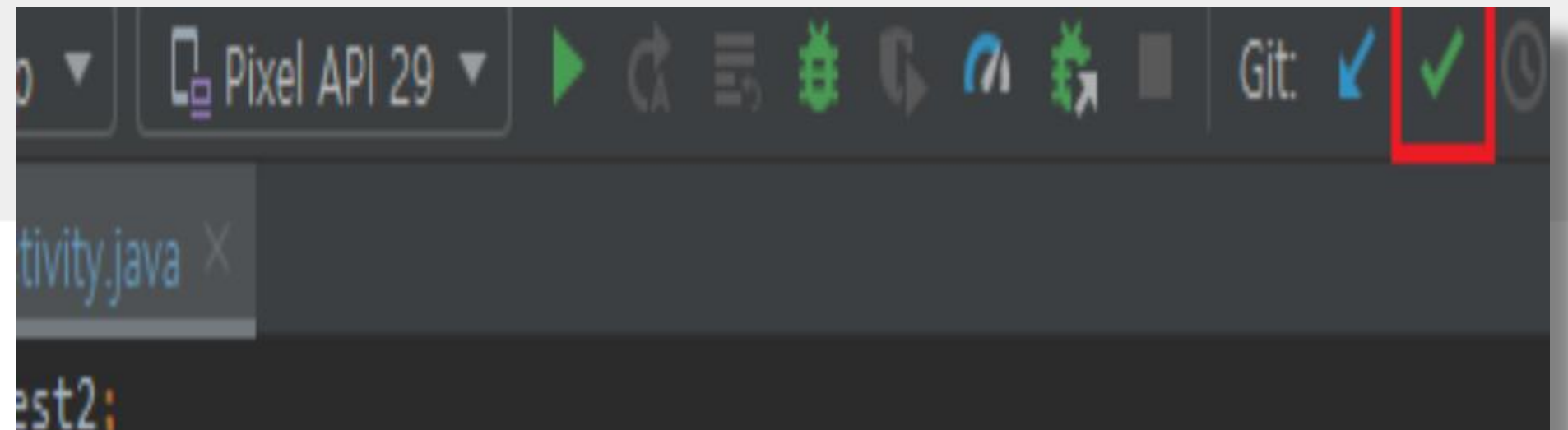
Git Hub

수정 프로젝트 업로드

프로젝트 수정 후 업데이트 시 방법

3가지 방법

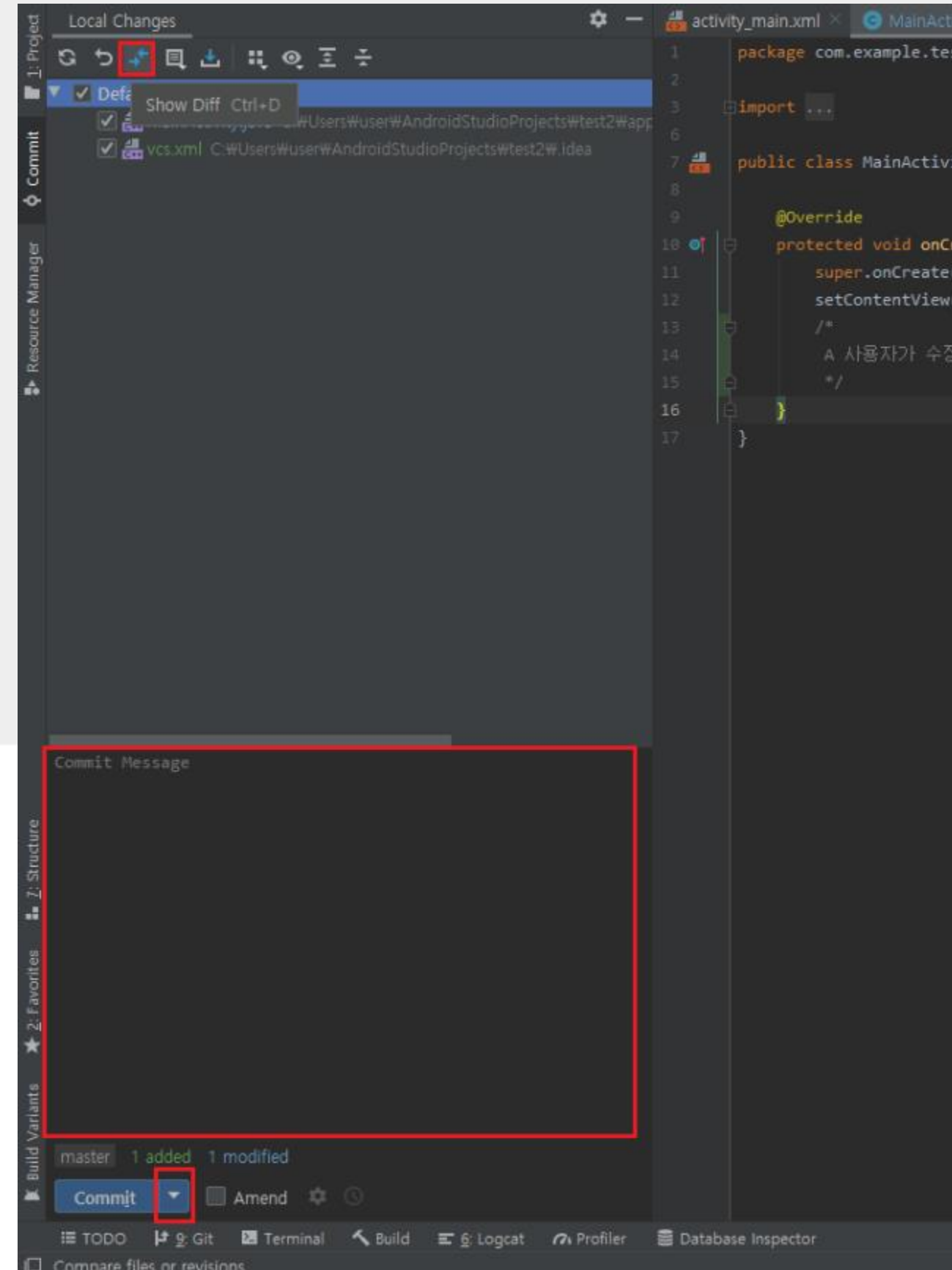
- 상단 Git - 체크 표시
- control + k 단축키
- 상단 메뉴 VCS - Git - Commit Files... 선택



Git Hub

수정한 프로젝트 업로드 (2)

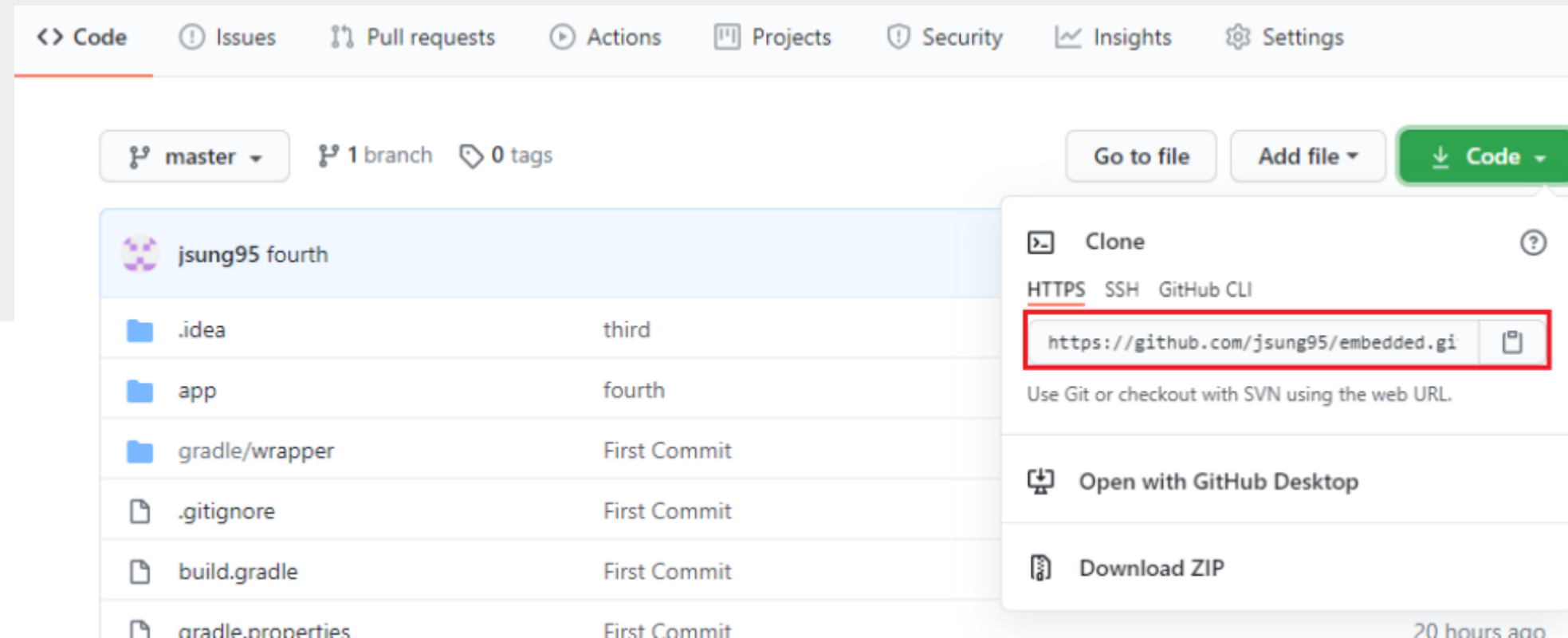
좌측 Commit 메뉴에서
Show Diff (Ctrl + D)를 통해 변경내용을 아주 보기 좋게 확인 가능
이후 Commit Message 입력 후
Commit 말고 Commit 우측 삼각 메뉴를 눌러 Commit and Push를 선택
이후 Push 해주면 깃허브 상에서 변경내용 확인 가능



Git Hub

협업

깃허브 초대 수락 후 Repository 내에서 Clone URL 복사
이후 마찬가지로 안드로이드 스튜디오에서 Git/GitHub 설정 후
첫 시작화면(Close Project로 프로젝트를 닫은 화면)에서
우측 Get from Version Control 메뉴 선택



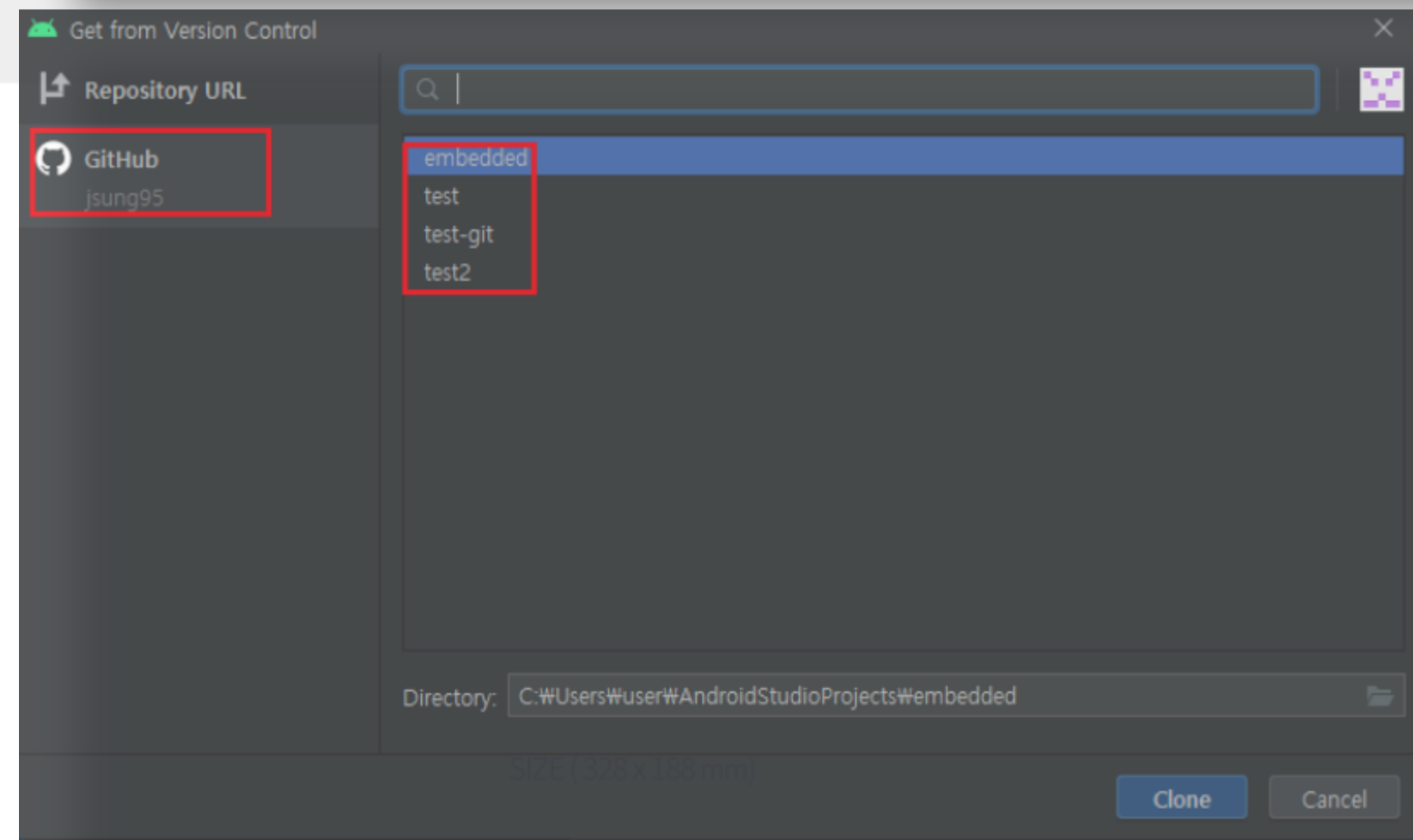
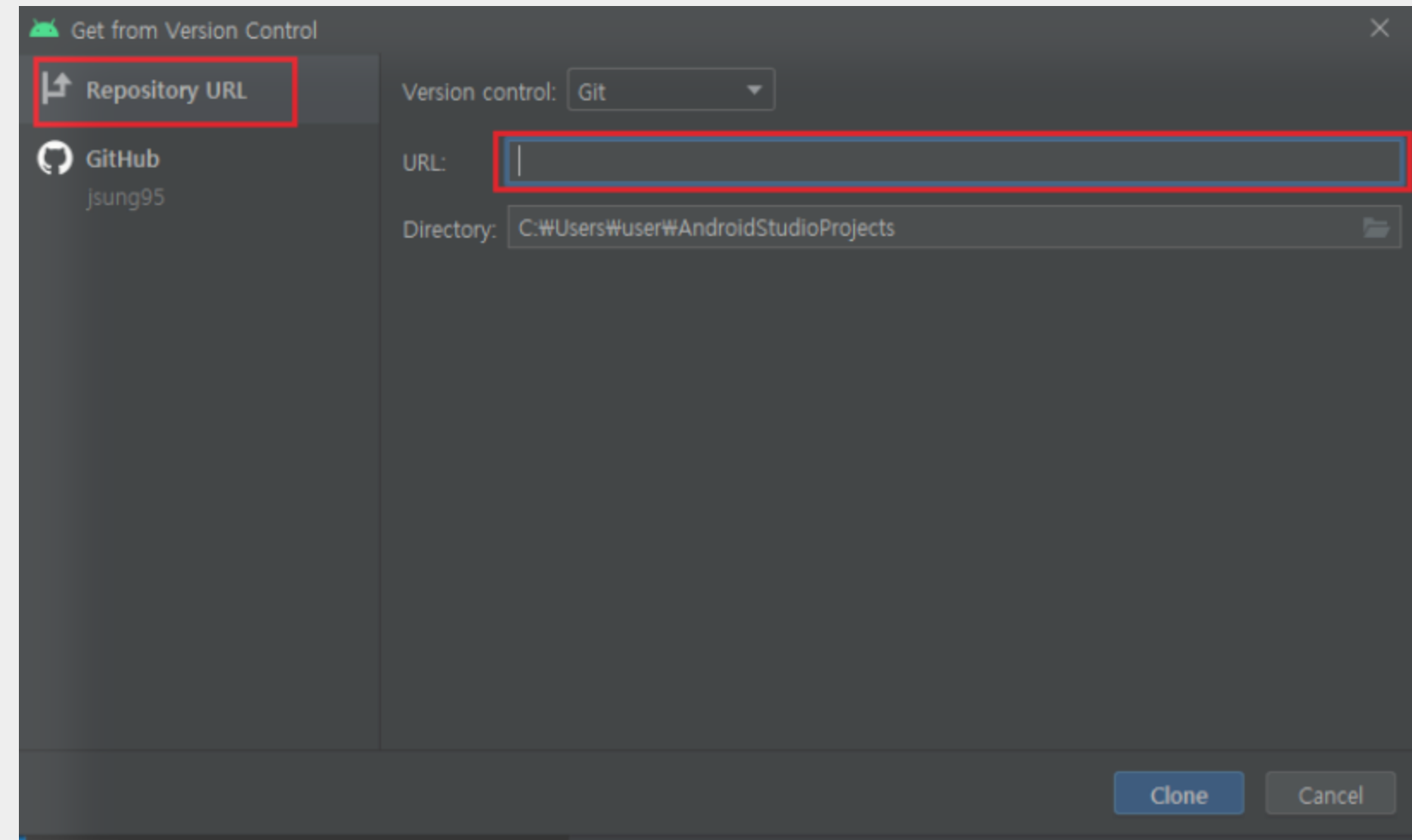
Git Hub

협업

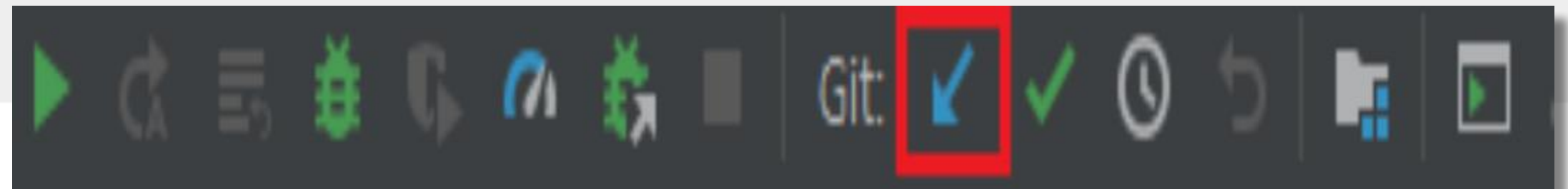
복사했던 Repository Clon URL 입력

또는

GitHub 탭에서 repository 선택 가능



Git Hub



협업


6. 협업자(Collaborator)가 코드 수정 후 → 업데이트 (Commit and Push)
협업자가 수정하던 프로젝트 Master가 수정하던 모두 마찬가지로
Commit and Push 로 깃허브에 업데이트

상단 Git - Update Project 를 통해 빠르게 업데이트
또는
상단 메뉴 VCS - Git - Pull 을 통해 업데이트 가능










Git Hub

협업 확인

- 1. 프로젝트를 공유한 협업자들의 Repository에 들어감
- 2. 우측 하단에 Contribution 에 나와있는 계정들이 협업자로 등록되어있는 사람들

 yelim0308 member_init edit

0155e4b 20 days ago 76 commits

 .idea	member_init edit	20 days ago
 app	member_init edit	20 days ago
 gradle/wrapper	Initial commit	2 months ago
 .gitignore	Initial commit	2 months ago
 build.gradle	rr	2 months ago
 gradle.properties	Initial commit	2 months ago
 gradlew	Initial commit	2 months ago
 gradlew.bat	Initial commit	2 months ago
 settings.gradle	rr	2 months ago

Help people interested in this repository understand your project by adding a README.

Add a README

No description, website, or topics provided.


Releases


No releases published
[Create a new release](#)


Packages

No packages published
[Publish your first package](#)

Contributors 3

 yelim0308

 hyun29570

 kuchic0918 KKS

Git Hub

협업 확인

프로젝트를 협업자들끼리 commit을하고 push를하게 되면 깃허브에 파일도 수정되어서 업로드가 된다. 추가로 언제 누가 커밋한지 볼 수도 있고 누가 어떤코드를 썼는지 알 수 있기 때문에 프로젝트간 필수로 해야하는 기능이다

yelim0308 member_init edit		
..		
AdapterComments.java	1109 12:44 kim	
AdapterCommunity.java	1109 12:44 kim	
CalendarFragment.java	.	
CommunityActivity.java	1109 12:44 kim	
CommunityDetailActivity.java	1109 12:44 kim	
CommunityFragment.java	1109 12:44 kim	
CommunityInfo.java	10/25 kks 22:17	
CommunityModel.java	1109 12:44 kim	
InfoFragment.java	member_init edit	
LoginActivity.java	1109 12:44 kim	
MainActivity.java	10/25 kks 22:17	
MainActivity2.java	1109 12:44 kim	
MapFragment.java	MapFragment add button(send step count to CalendarFragment)	
MemberInfo.java	11/02 23:07 KKS	
MemberInitActivity.java	1109 12:44 kim	
ModelComment.java	1109 12:44 kim	
PasswordResetActivity.java	modify layout	
SignUpActivity.java	1109 12:44 kim	
SplashActivity.java	11/02 23:07 KKS	
WalkFragment.java	.	

PROJECT.2

FIREBASE



02

-Firebase 란?

-Firebase 연동 법

FireBase

Firestore

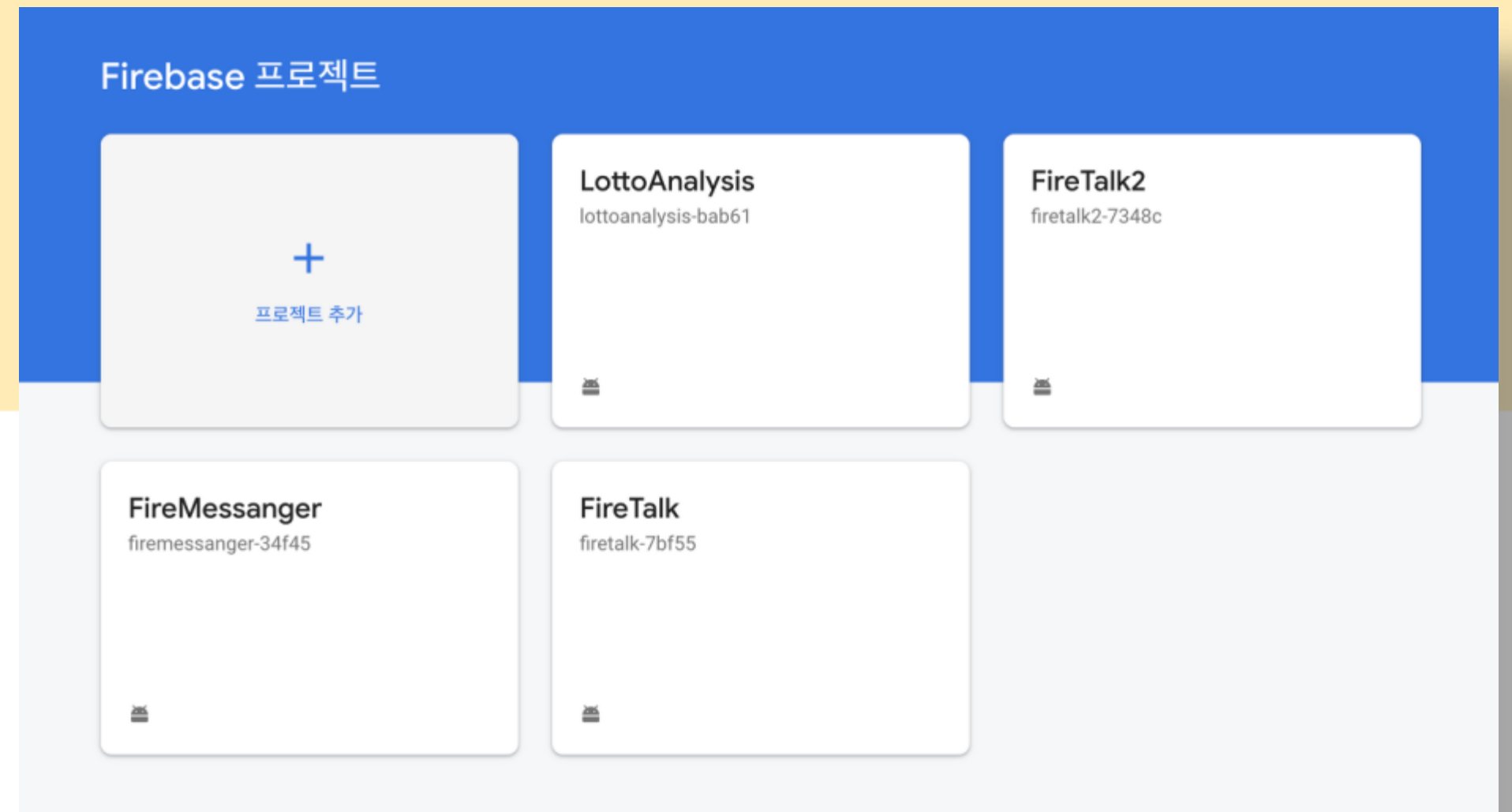
Firestore는 이 모든 플랫폼을 프로젝트 구축 시 자동적으로 만들어 줍니다. 또한 서버를 구축하기 위해서 리눅스 명령어를 알 필요도 없으며 도메인을 구입할 필요도 없고 개발하는 동안에는 서버를 구입할 필요도 없습니다.



FireBase

협업 확인

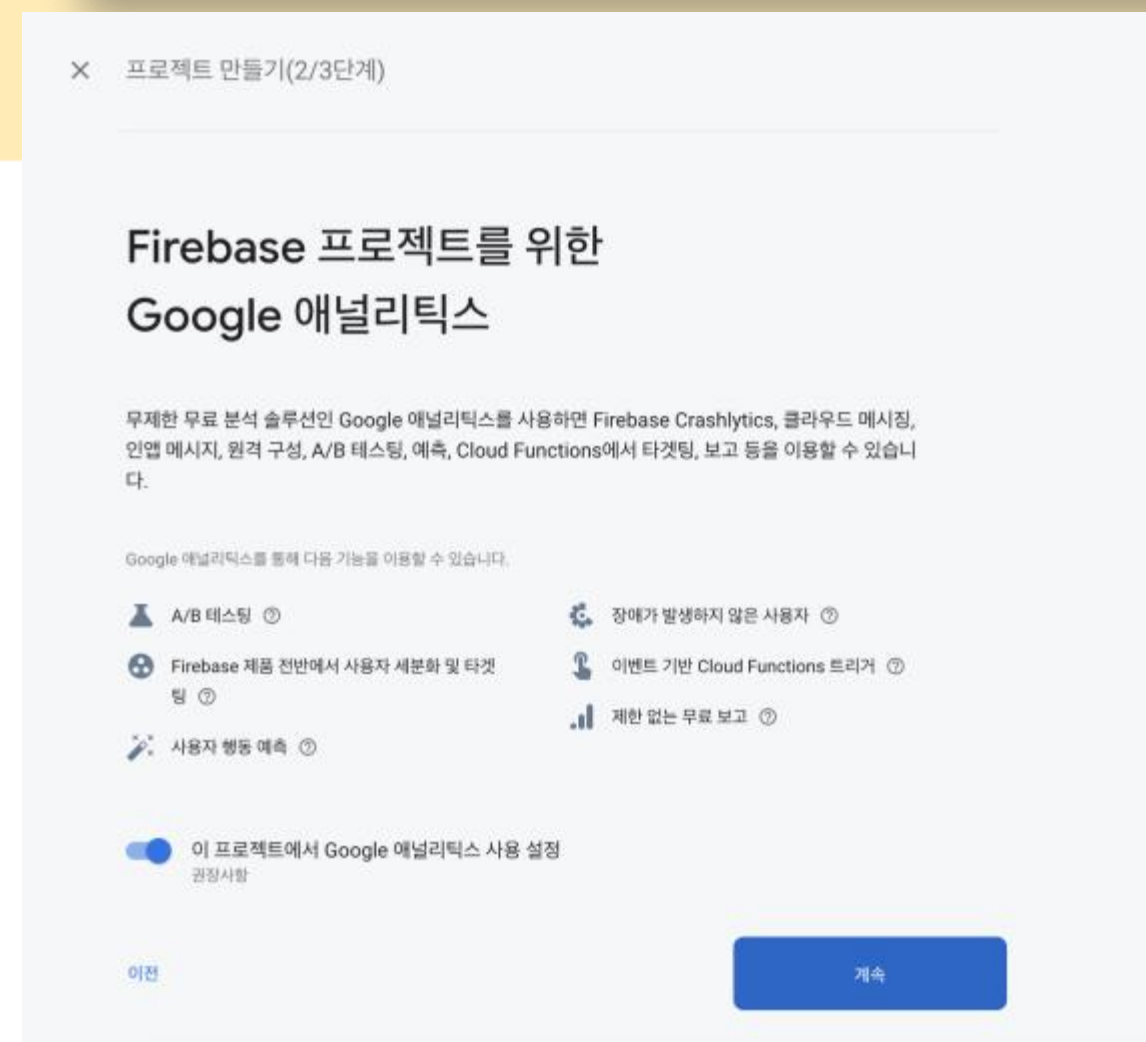
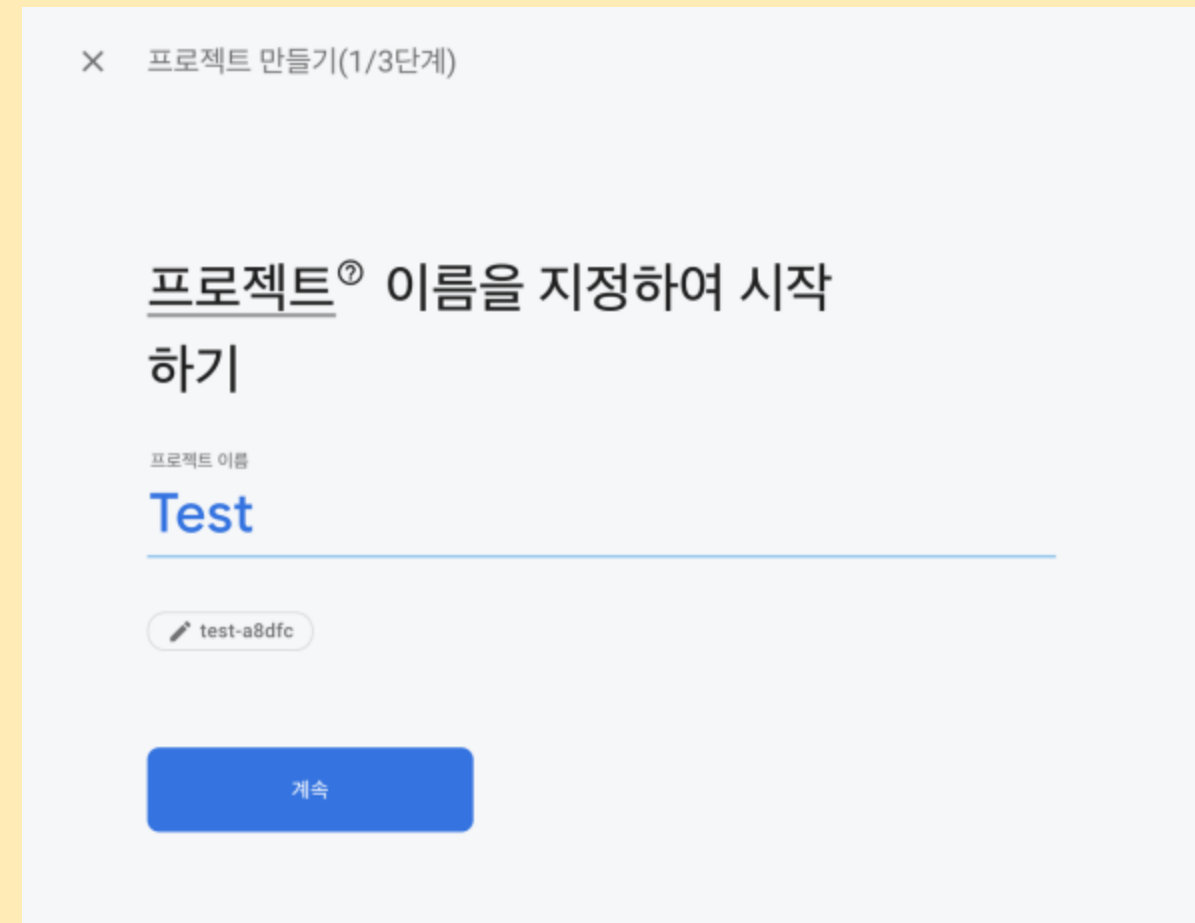
회원가입 후 로그인 하면 다음과 같은 화면이 나옴
(로그인은 구글로 연동 가능)
프로젝트 추가 버튼 누름



FireBase

협업 확인

프로젝트 이름을 지정한 후 "계속" 버튼을 클릭합니다.
Google 애널리틱스 를 사용할지를 정한 후 계속 버튼을 클릭합니다.




FireBase

× 프로젝트 만들기(3/3단계)

Google 애널리틱스 구성

Google 애널리틱스 계정 선택 또는 만들기 ⓘ

 Default Account for Firebase ▾

이 계정에서 자동으로 새 속성 만들기 ✎

프로젝트를 만들면 선택한 Google 애널리틱스 계정에 새 Google 애널리틱스 속성이 생성되고 Firebase 프로젝트에 연결됩니다. 이 연결을 통해 제품 간에 데이터 흐름이 활성화됩니다. Google 애널리틱스 속성에서 Firebase로 내보낸 데이터에는 Firebase 서비스 약관이 적용되지만 Google 애널리틱스로 가져온 Firebase 데이터에는 Google 애널리틱스 서비스 약관이 적용됩니다. [자세히 알아보기](#)

[이전](#)

프로젝트 만들기

협업 확인

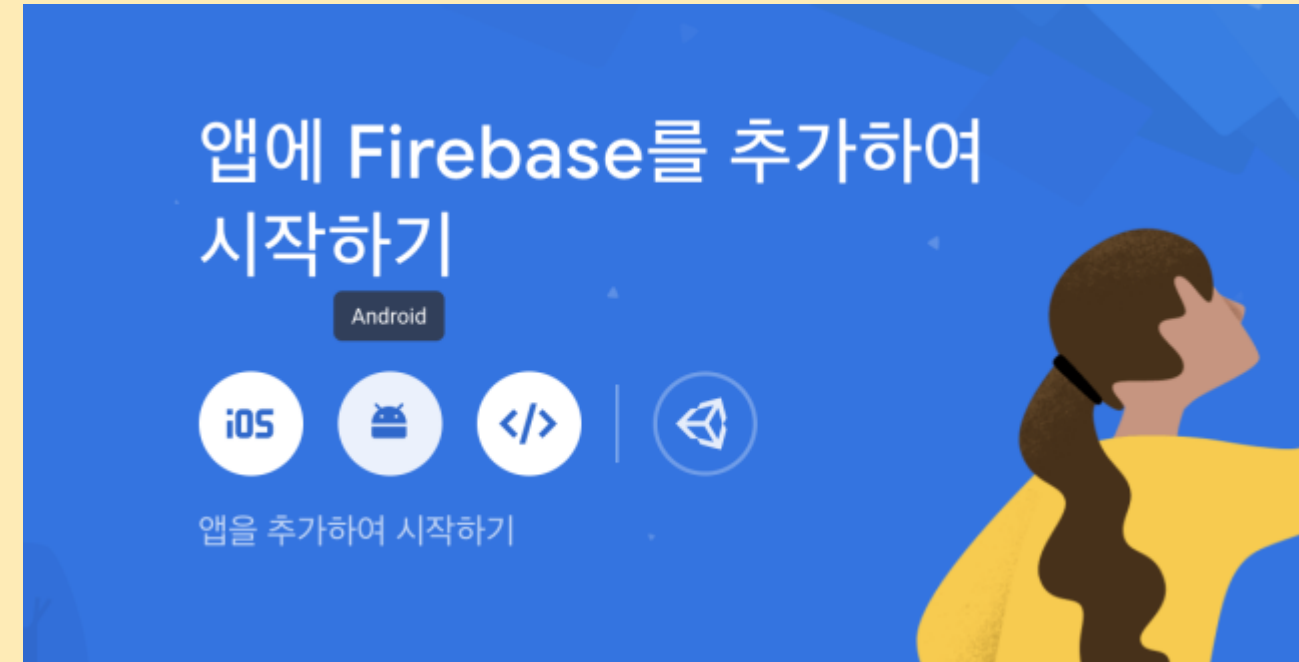
계정이 따로 없다면, Default Account for Firebase로 설정한 후 "프로젝트 만들기" 버튼을 클릭합니다.

프로젝트 생성 중 화면이 나오면 완료될 때 까지 기다립니다.



리소스 프로비저닝 중...
myYoutube

FireBase



협업 확인

프로젝트 창으로 이동하게 되는데, 여기서는 어떤 플랫폼에서 사용할지 지정할 수 있습니다.

안드로이드 모양 버튼을 클릭합니다.

Android 패키지 이름, 앱 닉네임(선택), 디버그 서명 인증서 SHA-1(선택)을 입력해야 합니다.

[illegible]

FireBase

× Android 앱에 Firebase 추가

1 앱 등록

Android 패키지 이름: com.Testcompany.test, 앱 닉네임: TestNick

2 구성 파일 다운로드

Android 스튜디오에 대한 안내(아래 참조) | [Unity](#) [C++](#)

↓ google-services.json 다운로드

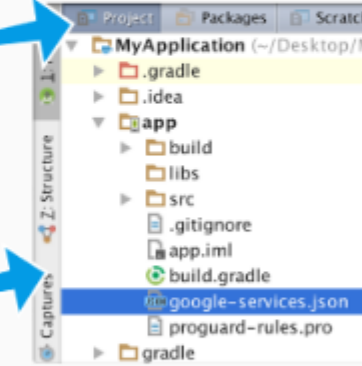
1

Android 스튜디오에서 프로젝트 보기로 전환하여 프로젝트 루트 디렉터리를 표시하세요.

방금 다운로드한 google-services.json 파일을 Android 앱 모듈 루트 디렉터리로 이동하세요.



google-services.json



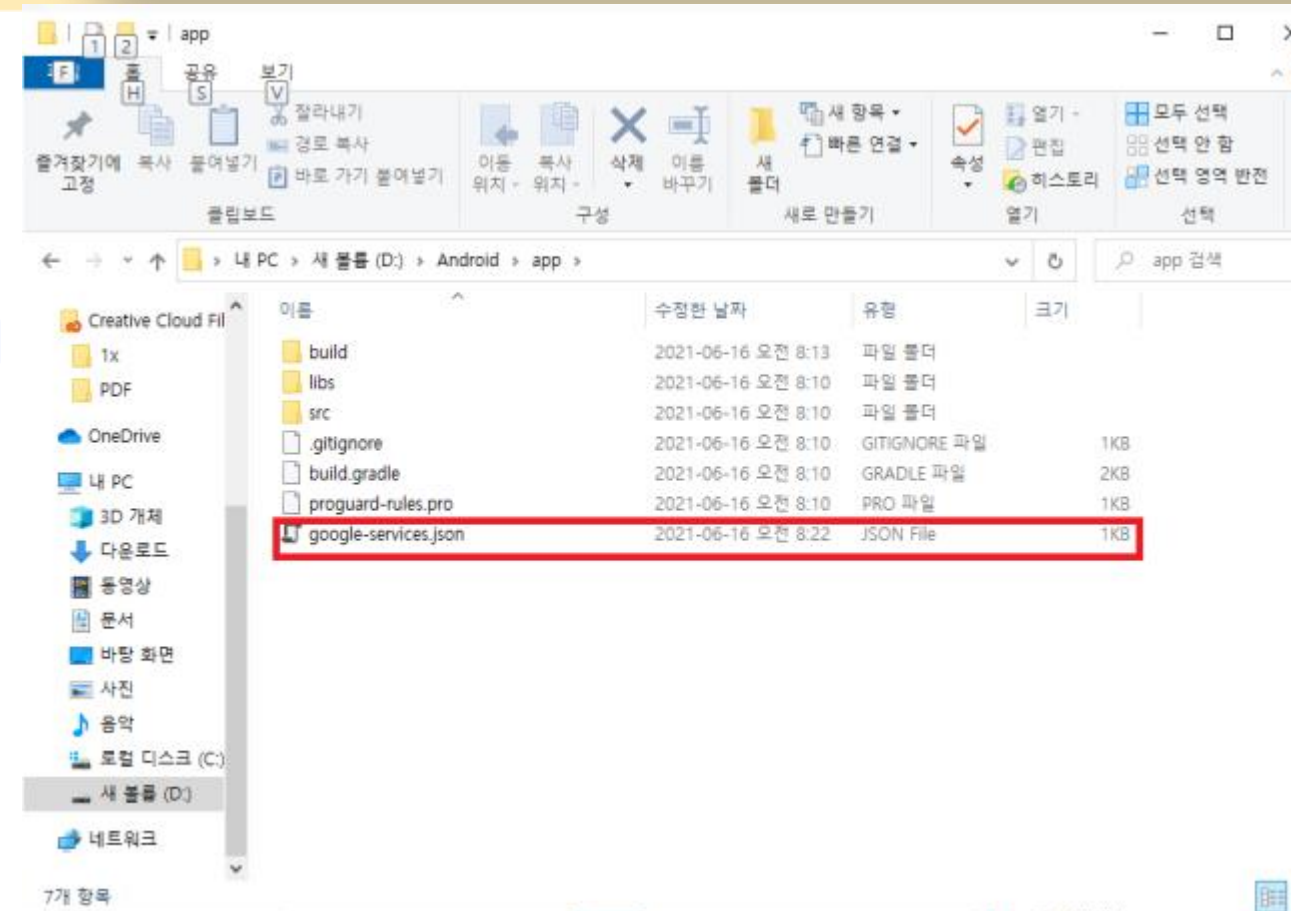
다음

2

3 Firebase SDK 추가

협업 확인

설명과 같이 다운로드 받은 google-services.json을 프로젝트의 app에 넣어줍니다.



FireBase

협업 확인

안드로이드 스튜디오와 Firebase를 연결을 해야합니다.

build.gradle파일을 수정해야합니다.


앱수준의 build.gradle, 프로젝트 수준의 build.gradle을 수정해야합니다.

밑에 이미지에서는 1번이 앱수준의 build.gradle, 2번이 프로젝트 수준의 build.gradle입니다.

3

Firebase SDK 추가

Gradle 안내 | [Unity](#) [C++](#)

[Gradle](#)  용 Google 서비스 플러그인에서 방금 다운로드한 google-services.json 파일을 로드합니다. build.gradle 파일을 수정하여 플러그인을 사용하세요.

프로젝트 수준의 build.gradle (<project>/build.gradle):

```
buildscript {
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
    }
    dependencies {
        ...
        // Add this line
        classpath 'com.google.gms:google-services:4.3.3'
    }
}

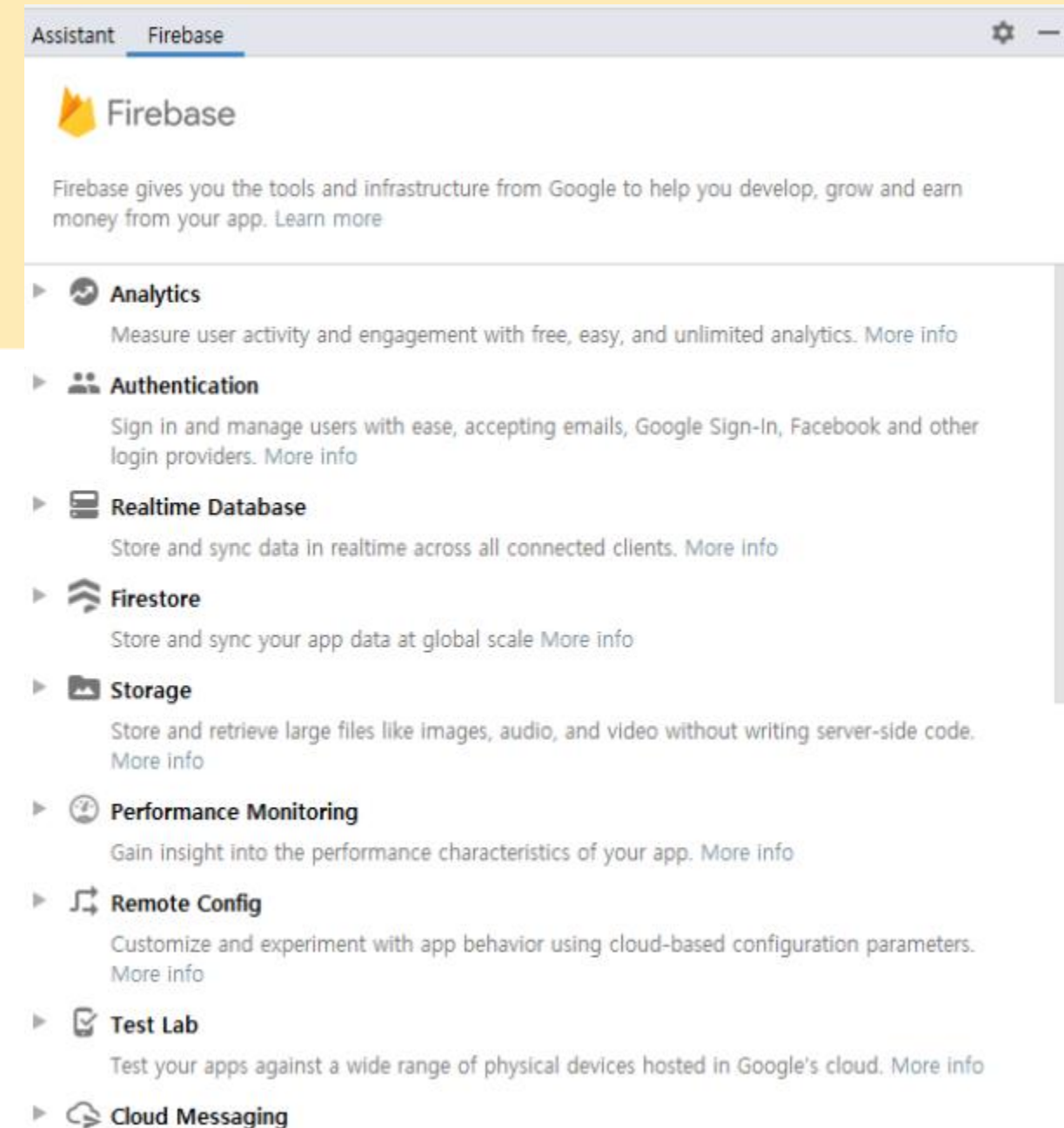
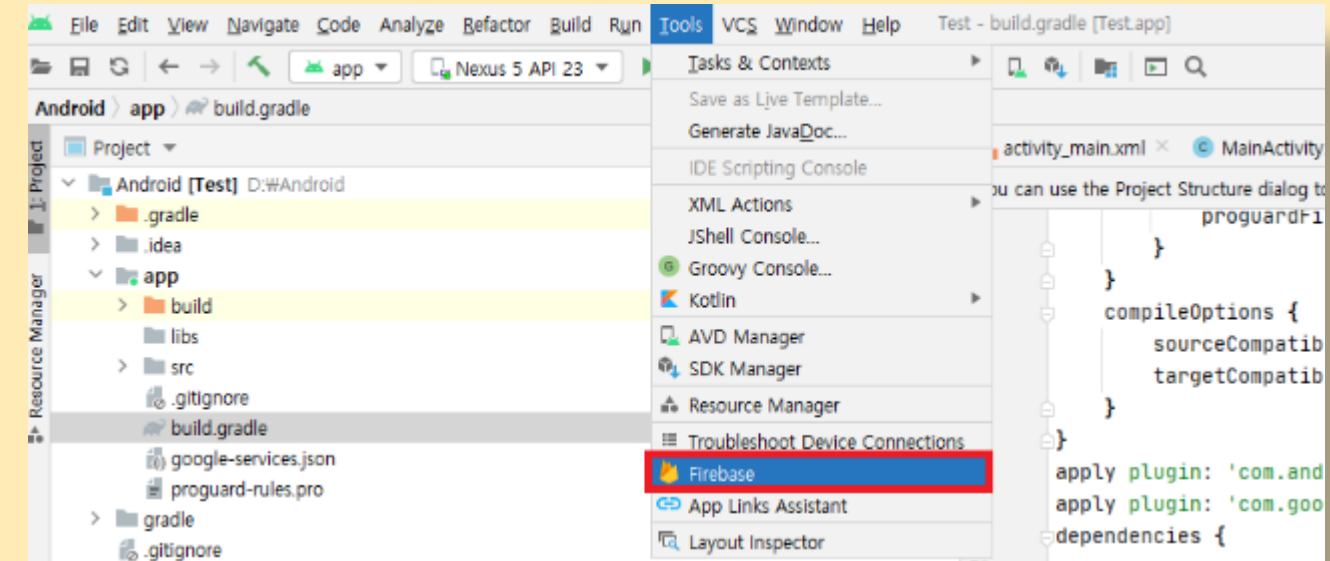
allprojects {
    ...
    repositories {
        // Check that you have the following line (if not, add it):
        google() // Google's Maven repository
        ...
    }
}
```

앱 수준의 build.gradle (<project>/<app-module>/build.gradle):

```
apply plugin: 'com.android.application'
// Add this line
apply plugin: 'com.google.gms.google-services'

dependencies {
    // add the Firebase SDK for Google Analytics
    implementation 'com.google.firebase:firebase-analytics:17.2.2'
    // add SDKs for any other desired Firebase products
    // https://firebase.google.com/docs/android/setup#available-libraries
}
```

FireBase



협업 확인

연결을 확인합니다.

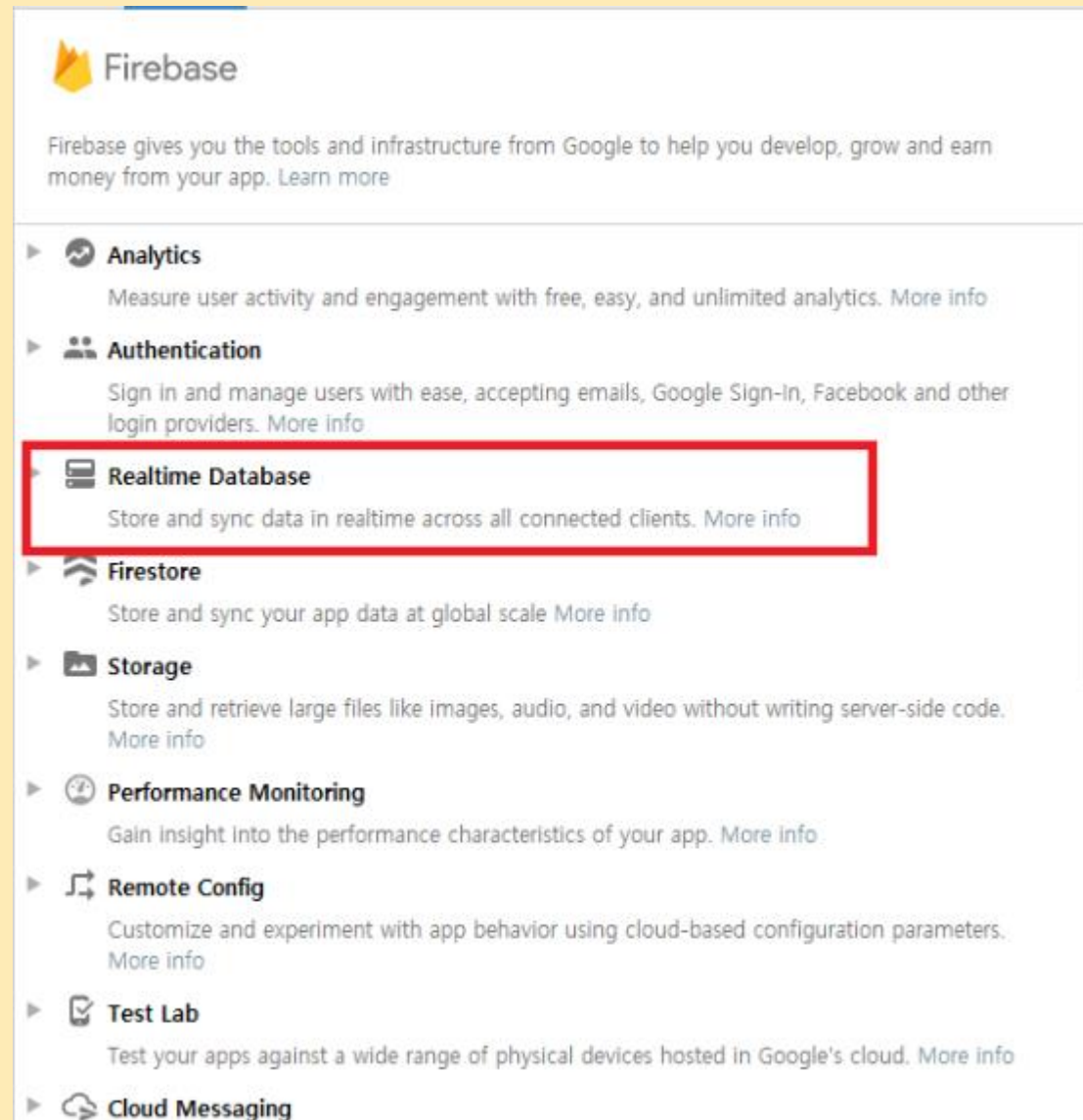
안드로이드 스튜디오 메뉴에서 Tool-Firebase를 클릭합니다.

Firebase 메뉴가 나옵니다.

저는 실시간 디비를 통해 연결을 확인해보도록 하겠습니다.

3번째 Realtime Database를 클릭해봅니다.

FireBase



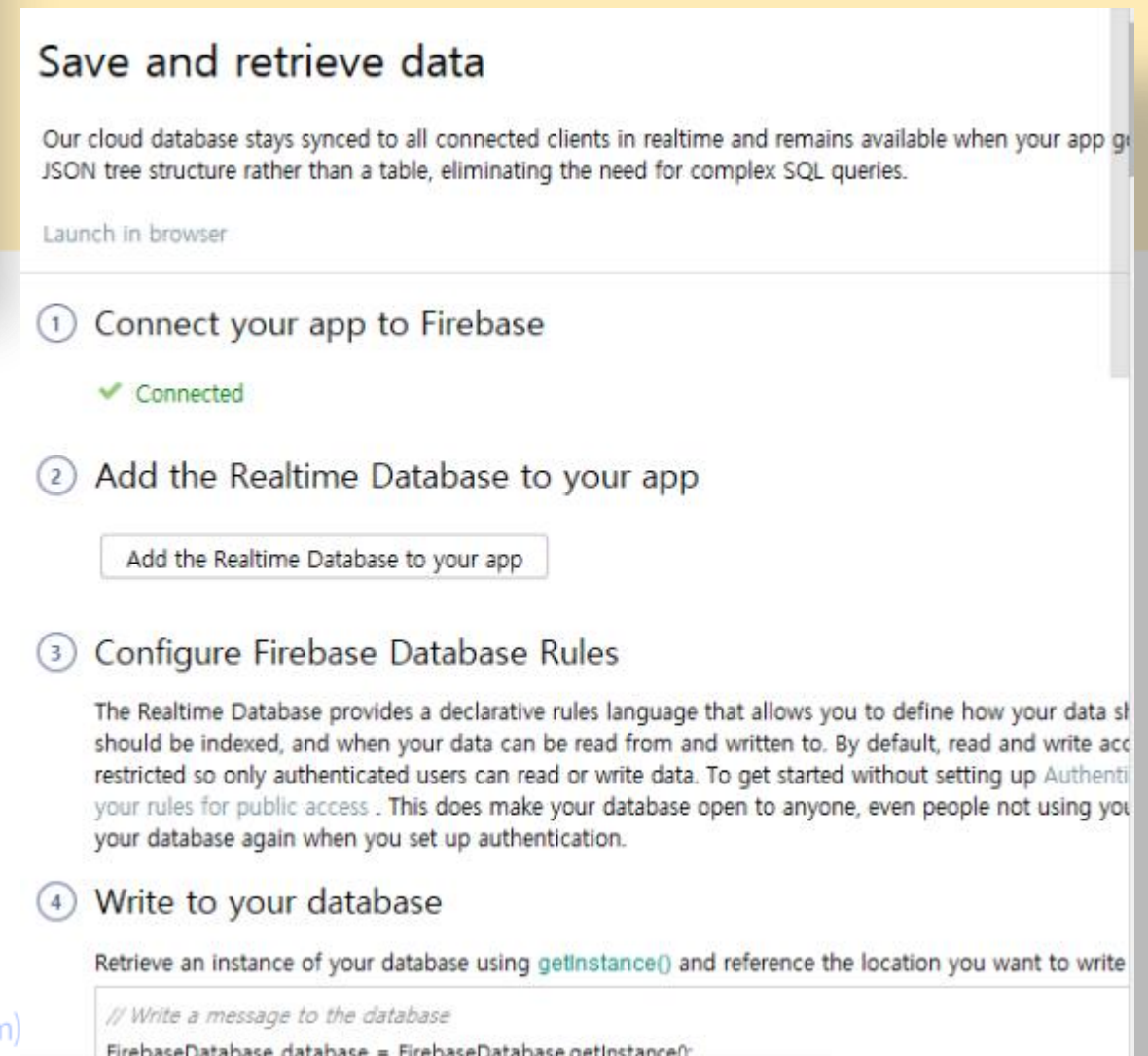
협업 확인

Save and retrieve data가 생깁니다. 클릭합니다.

다음과 같은 화면이 나옵니다.

1, 2번이 모두 체크가 되어야 합니다

2번은 아직 체크가 안되었기 때문에 Add to Realtime Database to your app을 클릭합니다.

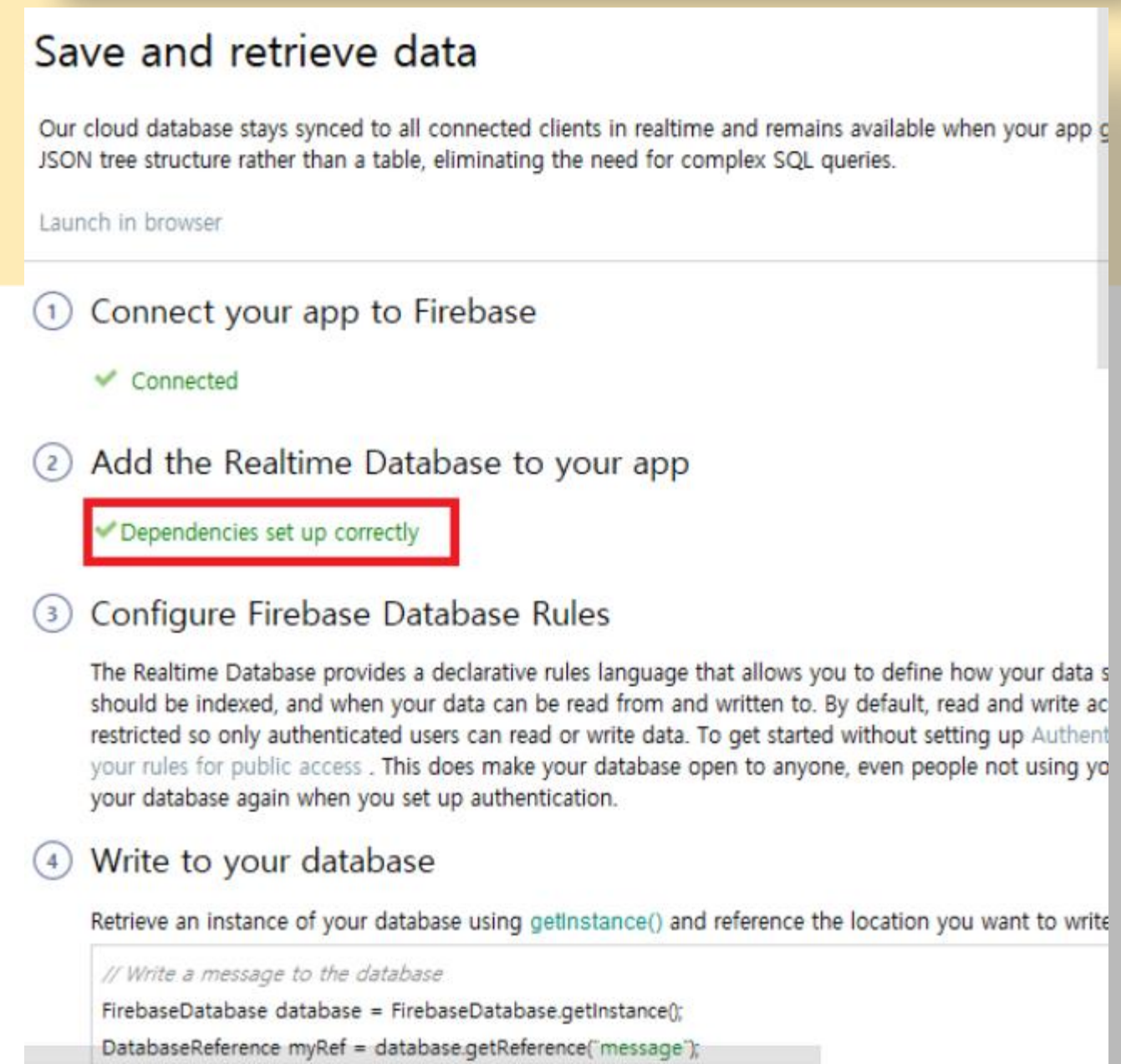
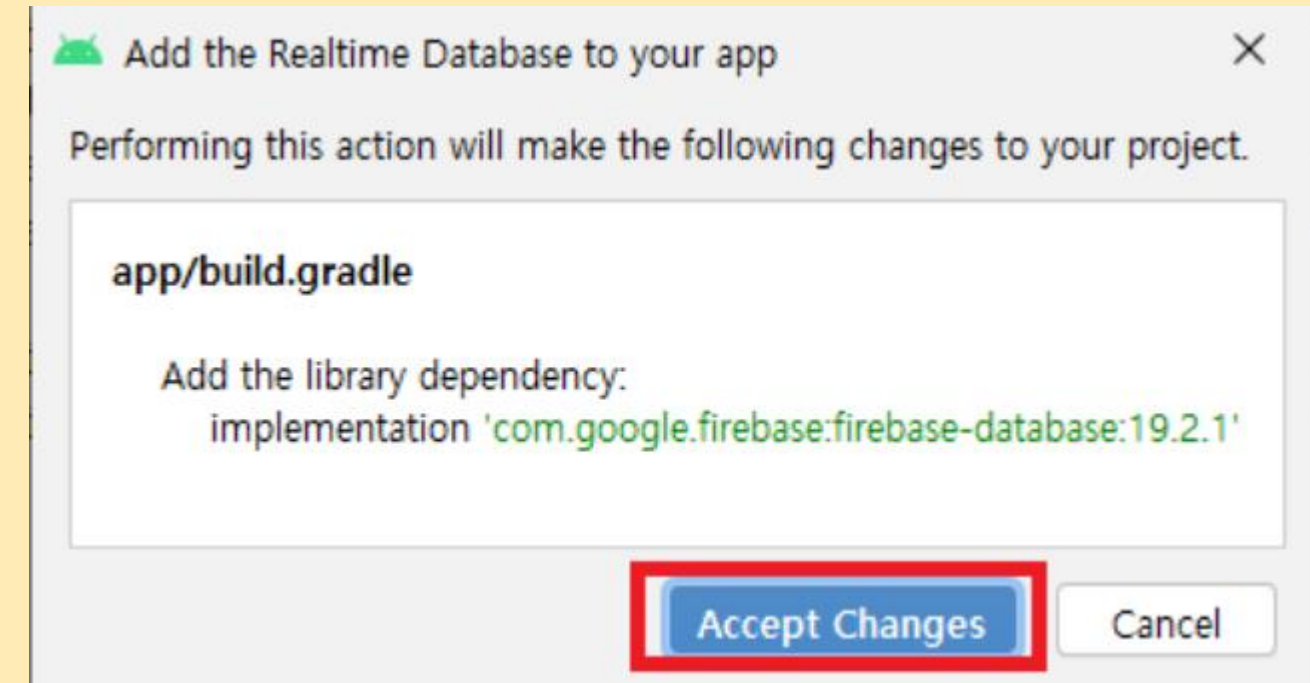


SIZE (328 x 188 mm)

FireBase

협업 확인

클릭하면 메시지 박스가 하나 뜹니다. app/build.gradle파일에 database 관련 라이브러리 관련 구문을 추가하겠냐고 물어봅니다. Accept Changes를 클릭합니다. 잠시 기다리면 "Dependencies set up correctly"로 변경된 것을 볼 수 있습니다.



PROJECT.3

SNS



03

-UI/UX

-기능 설명-

회원가입 / 로그인페이지 (1)

```
private void signUp() {
    String email = ((EditText) findViewById(R.id.edittext)).getText().toString();
    String password = ((EditText) findViewById(R.id.passwordtext)).getText().toString();
    String passwordcheck = ((EditText) findViewById(R.id.passwordcheck)).getText().toString();

    if(email.length() > 0 && password.length() > 0 && passwordcheck.length() > 0){
        if(password.equals(passwordcheck)){
            mAuth.createUserWithEmailAndPassword(email, password)
                .addOnCompleteListener(activity, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            FirebaseUser user = mAuth.getCurrentUser();
                            String email = user.getEmail();
                            String uid = user.getUid();

                            HashMap<Object, String> hashMap = new HashMap<>();

                            hashMap.put("email", email);
                            hashMap.put("uid", uid);
                            hashMap.put("name", "");
                            hashMap.put("image", "");

                            FirebaseDatabase database = FirebaseDatabase.getInstance();
                            //
                            DatabaseReference reference = database.getReference("Users");

                            reference.child(uid).setValue(hashMap);

                            showToast(msg: "회원가입에 성공하였습니다.");
                            myStartActivity(MemberInitActivity.class);
                        } else {
                            if(task.getException() != null){
                                showToast(task.getException().toString());
                            }
                        }
                    }
                });
        } else {
            showToast(msg: "비밀번호가 일치하지 않습니다.");
        }
    } else {
        showToast(msg: "이메일 또는 비밀번호를 입력해 주세요.");
    }
}
```

SKT 3:03 40%

회원 가입

이메일

비밀번호

비밀번호 확인

회원가입

로그인

Firebase에 저장된 회원

이메일 주소, 전화번호 또는 사용자 UID로 검색	사용자 추가			
식별자	제공업체	생성된 날짜 ↓	로그인한 날짜	사용자 UID
rabbit010308@naver.com		2021. 11. 9.	2021. 11. 9.	dnxNXPSvsRpSMi08RaZ4a9FC6...
test@naver.com		2021. 11. 9.	2021. 11. 9.	QZ9FtmjhgBvVcqu5qE85mL8NGK...
test0@naver.com		2021. 11. 9.	2021. 11. 9.	OeSS3qBD8MXM8fchs8eD8u2CP...
test00@naver.com		2021. 11. 9.	2021. 11. 9.	HeMGYmbkRLX6fjgK3g892XVg7H2...
kks980918@naver.com		2021. 11. 9.	2021. 11. 9.	:SYL1DxXbHmZg58bBJRJt0ZAKhQ...
rabbit010308@naver.com		2021. 11. 9.	2021. 11. 9.	5XELgcHXQmFu6DFHU78Dytq54...
rabbit010308@naver.com		2021. 11. 9.	2021. 11. 9.	O6uV6Ko5VmKUE54JN3JC2HJr...
test11@naver.com		2021. 11. 9.	2021. 11. 9.	0sFRQAJjwhfdwnyulua392VaRbi2
kuchic0918@naver.com		2021. 11. 9.	2021. 11. 9.	c33FmAMSp6iVF9baH8eo4j6cf2
shyun29570@gmail.com		2021. 11. 4.	2021. 11. 4.	g0Pgkyze9yfkaoK1nYkNHLfmyz1
rabbit010308@gmail.com		2021. 11. 4.	2021. 11. 9.	tnZFq6i2dfn6e531mD9McJhZ9j2
kks980918@gmail.com		2021. 11. 1.	2021. 11. 29.	r5zVpENpB7PmmxcjzMzz9FVU1G2

회원가입 / 로그인페이지 (1)

Google 로그인에 필요한 코드

```
private void login() {
    String email = ((EditText)findViewById(R.id.emailEditText)).getText().toString();
    String password = ((EditText)findViewById(R.id.passwordEditText)).getText().toString();

    if(email.length() > 0 && password.length() > 0){
        auth.signInWithEmailAndPassword(email, password)
            .addOnCompleteListener(activity, this, new OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if (task.isSuccessful()) {
                        FirebaseUser user = auth.getCurrentUser();
                        String email = user.getEmail();
                        String uid = user.getUid();

                        HashMap<Object, String> hashMap = new HashMap<>();

                        hashMap.put("email", email);
                        hashMap.put("uid", uid);
                        hashMap.put("name", "");
                        hashMap.put("image", "");

                        FirebaseDatabase database = FirebaseDatabase.getInstance();
                        //
                        DatabaseReference reference = database.getReference(path: "Users");

                        reference.child(uid).setValue(hashMap);

                        showToast(msg: "로그인에 성공하였습니다.");
                        myStartActivity(MainActivity2.class);
                    } else {
                        if(task.getException() != null){
                            showToast(task.getException().toString());
                        }
                    }
                }
            });
    } else {
        showToast(msg: "이메일 또는 비밀번호를 입력해 주세요.");
    }
}
```



```
GoogleSignInOptions googleSignInOptions = new GoogleSignInOptions.Builder(GoogleSignInOptions.DEFAULT_SIGN_IN)
    .requestIdToken("777276898973-ahjdg0f37tcu142gdm5vht71i04r04b.apps.googleusercontent.com")
    .requestEmail()
    .build();

googleApiClient = new GoogleApiClient.Builder(context).enableAutoManage(fragmentActivity, this, onConnectionFailedListener: this)
    .addApi(Auth.GOOGLE_SIGN_IN_API, googleSignInOptions)
    .build();

auth = FirebaseAuth.getInstance();

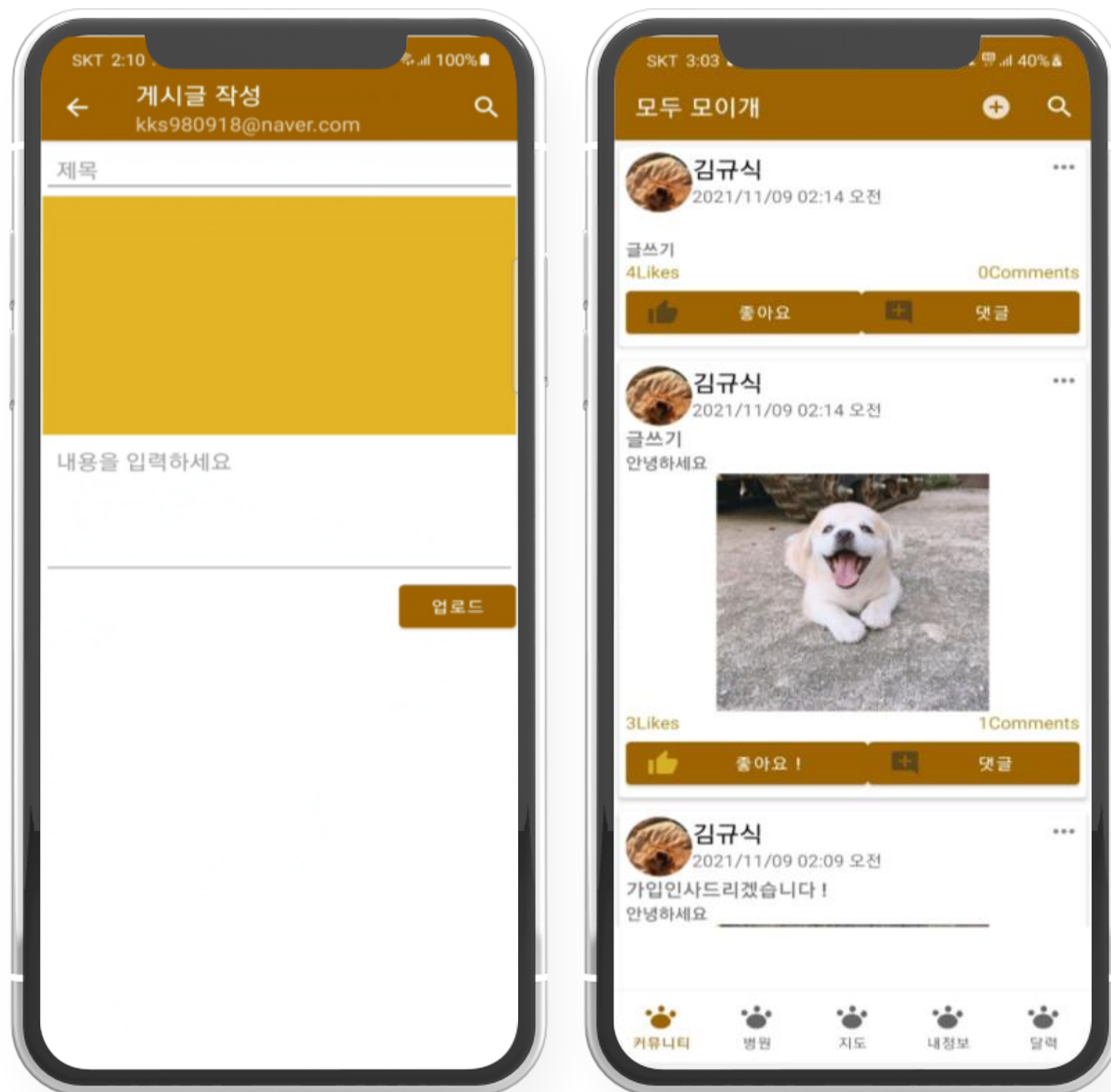
btn_google = findViewById(R.id.btn_google);
btn_google.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = Auth.GoogleSignInApi.getSignInIntent(googleApiClient);
        startActivityForResult(intent, REQ_SIGN_GOOGLE);
    }
});
```

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);

    if (requestCode == REQ_SIGN_GOOGLE) {
        GoogleSignInResult result = Auth.GoogleSignInApi.getSignInResultFromIntent(data);
        if (result.isSuccess()) {
            GoogleSignInAccount account = result.getSignInAccount();
            resultLogin(account);
        }
    }
}

private void resultLogin(GoogleSignInAccount account) {
    AuthCredential credential = GoogleAuthProvider.getCredential(account.getIdToken(), accessToken: null);
    auth.signInWithCredential(credential).addOnCompleteListener(activity, this, new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            if (task.isSuccessful()) {
                Toast.makeText(context, LoginActivity.this, text: "구글 로그인 성공", Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(getApplicationContext(), MainActivity2.class);
                intent.putExtra("name", "nickName", account.getDisplayName());

                startActivity(intent);
            } else {
                Toast.makeText(context, LoginActivity.this, text: "로그인 실패", Toast.LENGTH_SHORT).show();
            }
        }
    });
}
```



UI/UX / 기능

메인 화면

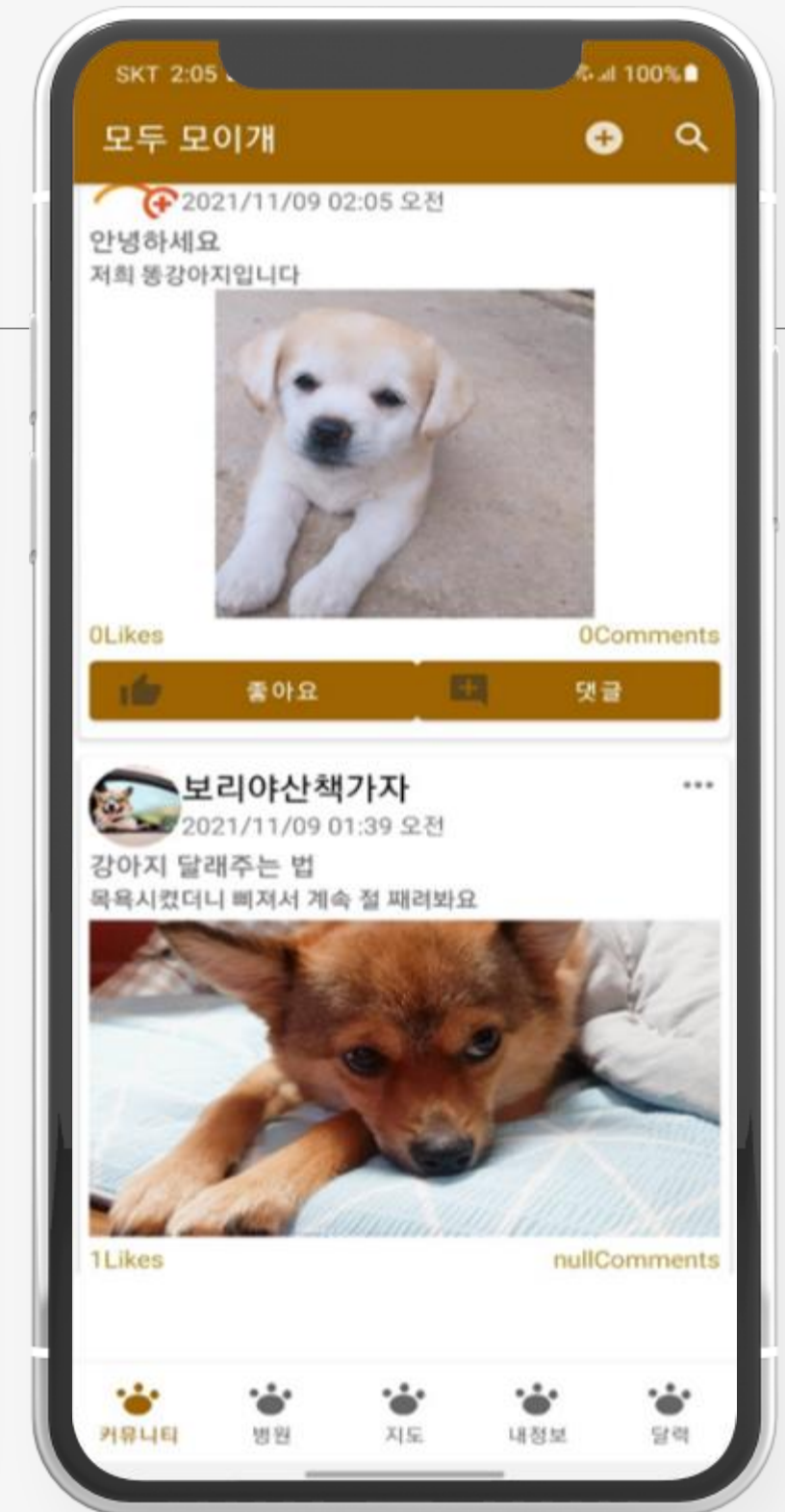
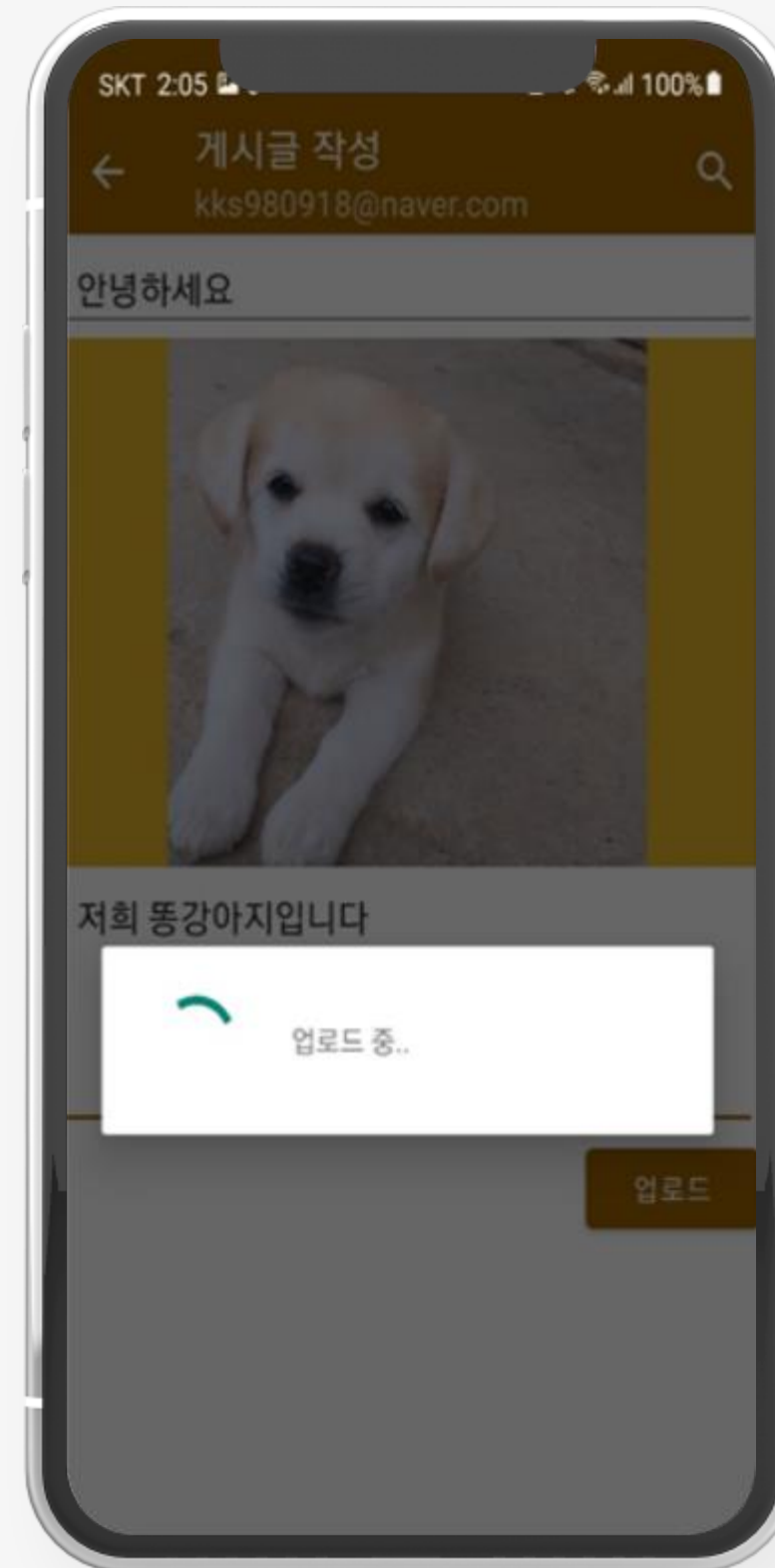
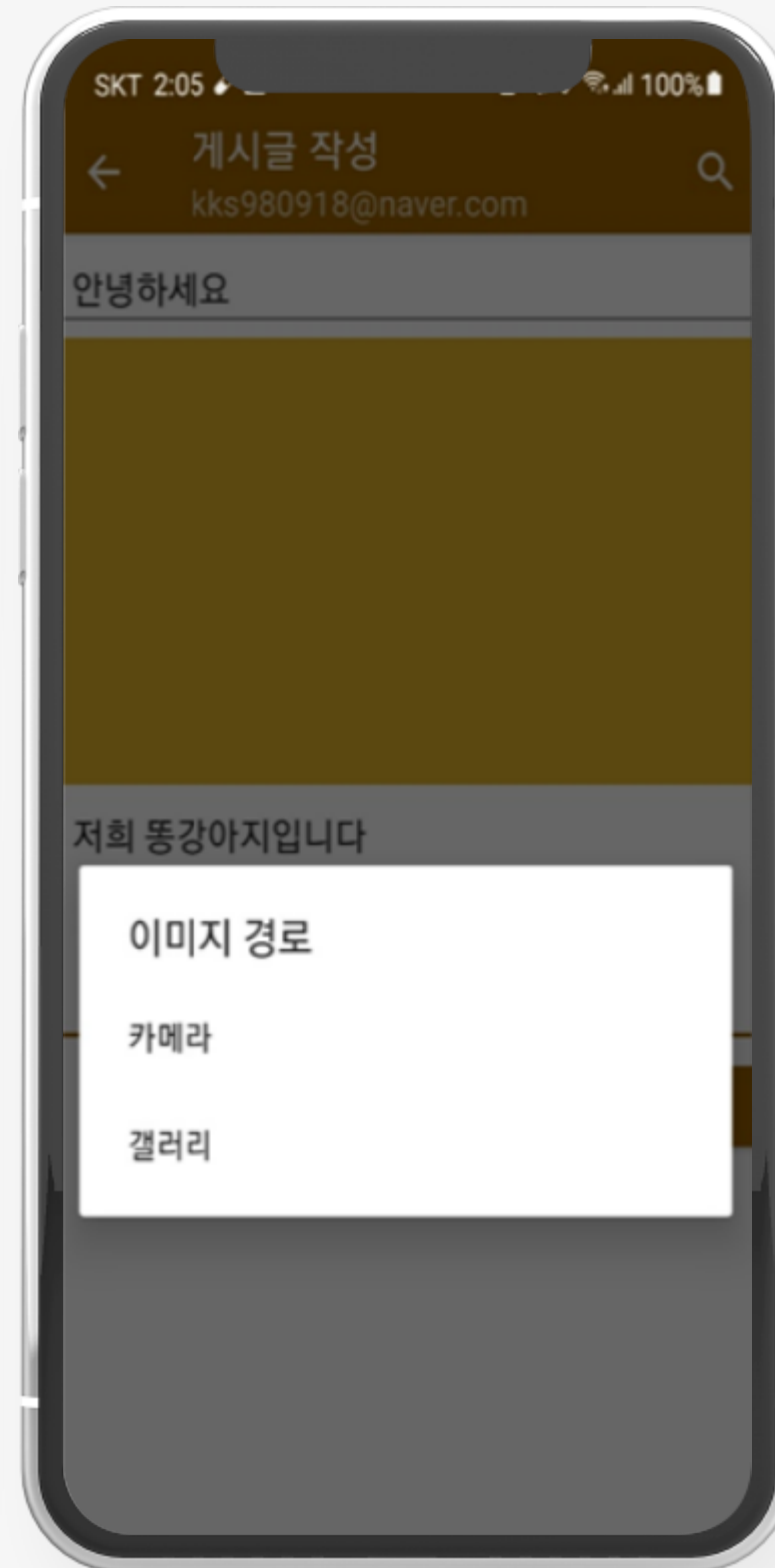
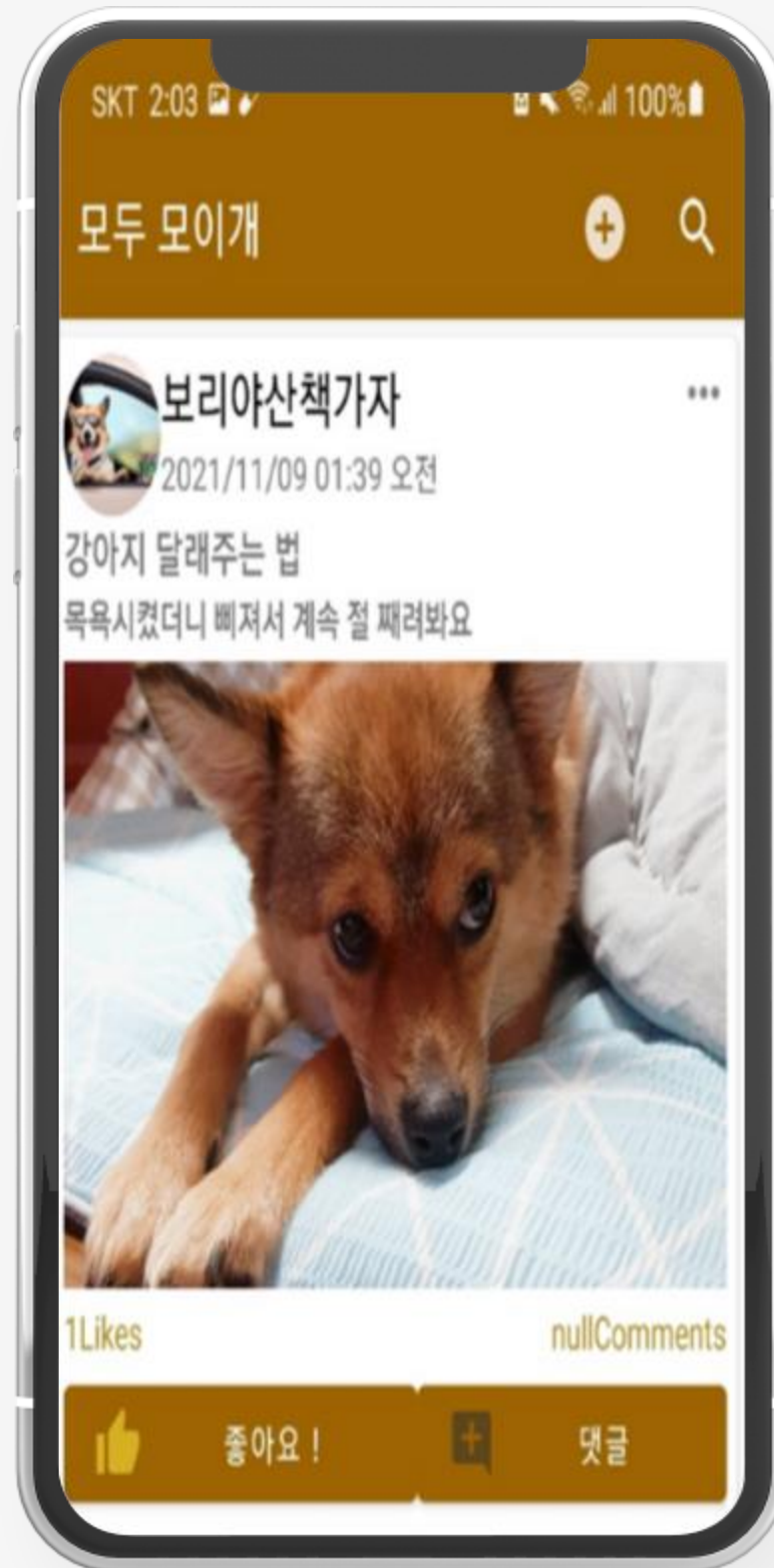
- DB에 저장된 이름, Timestamp로 기록한 시간, 저장된 프로필 사진을 받아서 글에 첨부되게 한다
- 글은 사진을 첨부할 수도있고 글만 쓸 수도 있다 대신 둘다 공백일 경우 업로드가 되지 않는다.

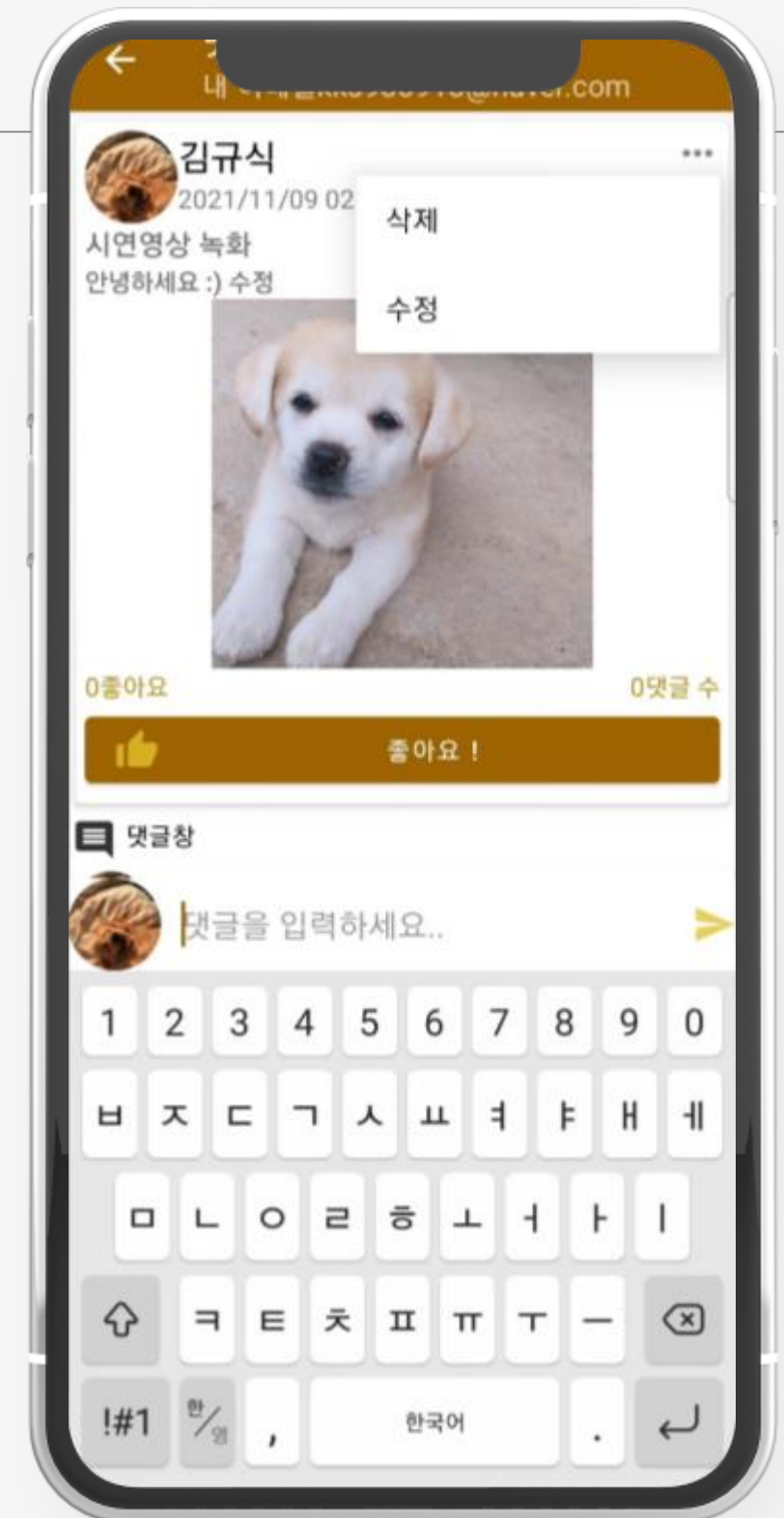
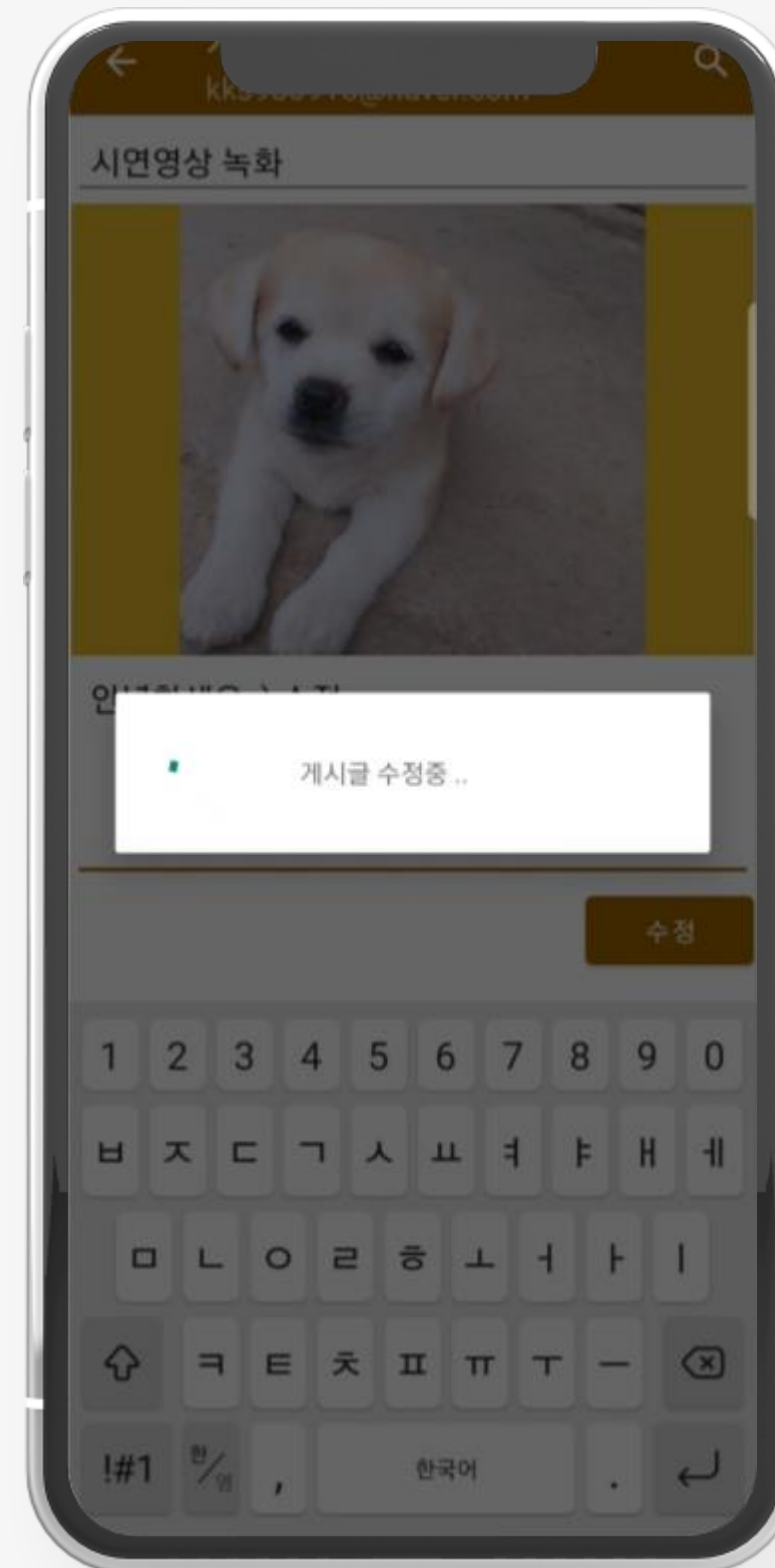
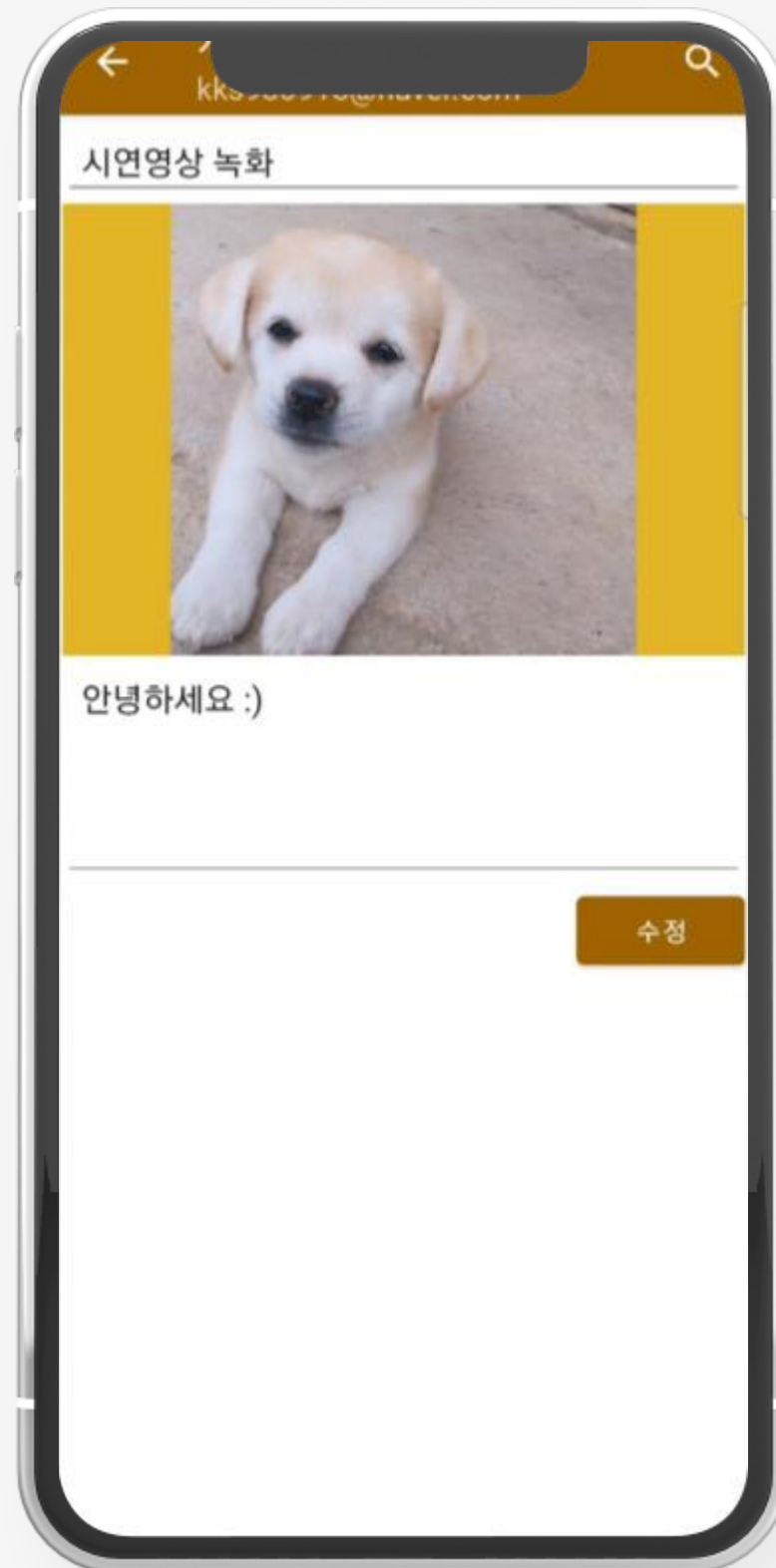
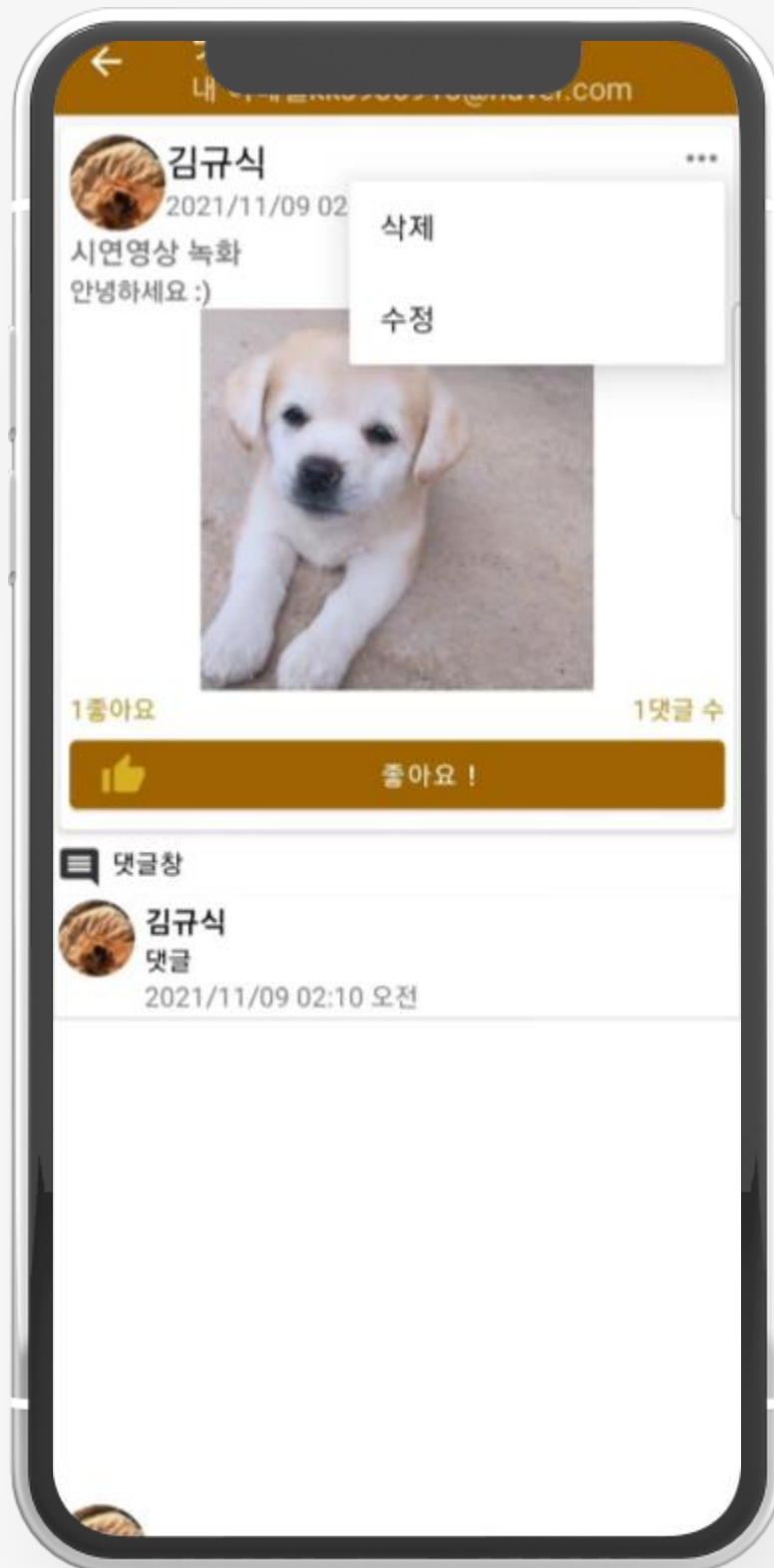
글쓰기 화면

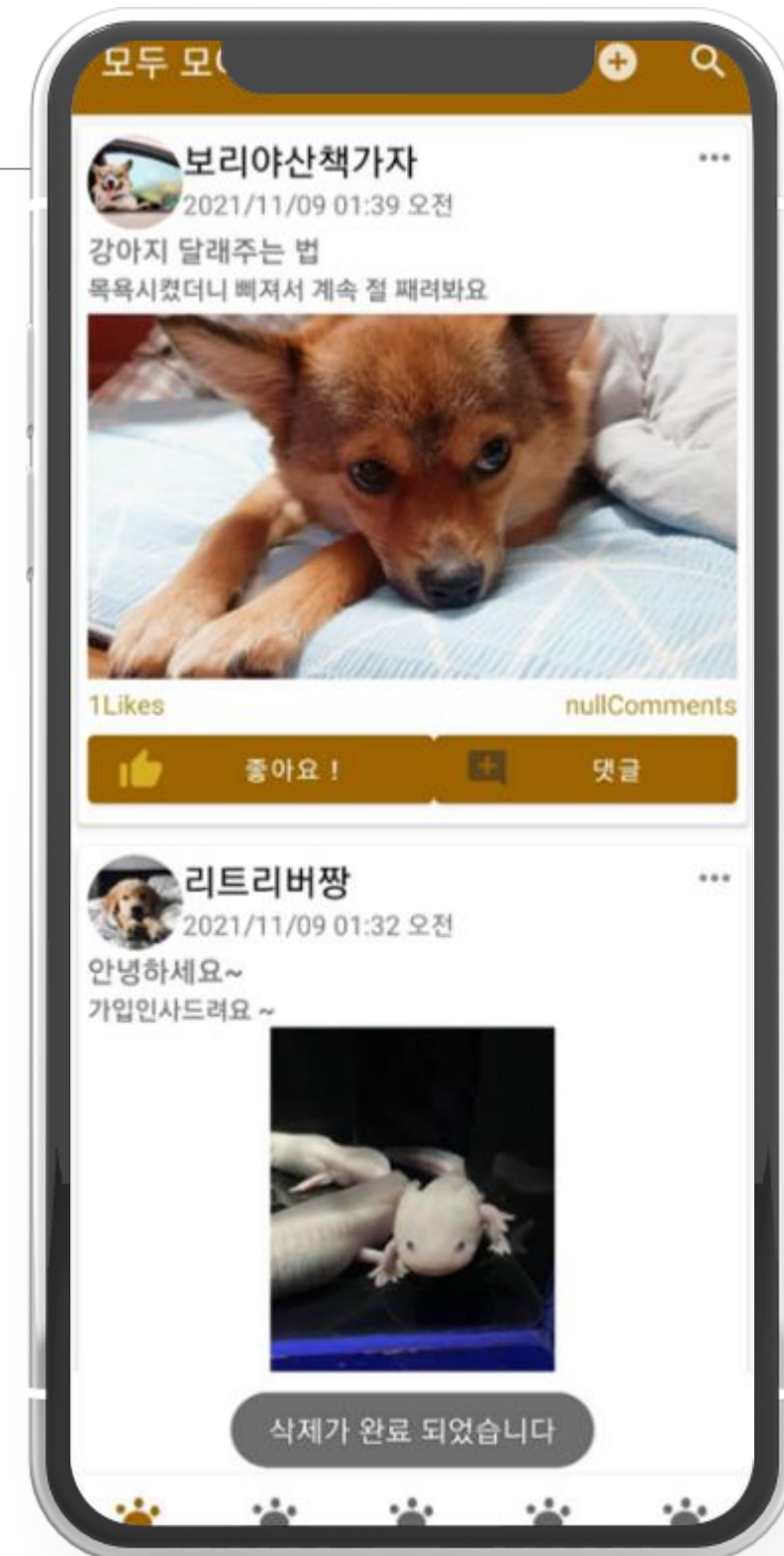
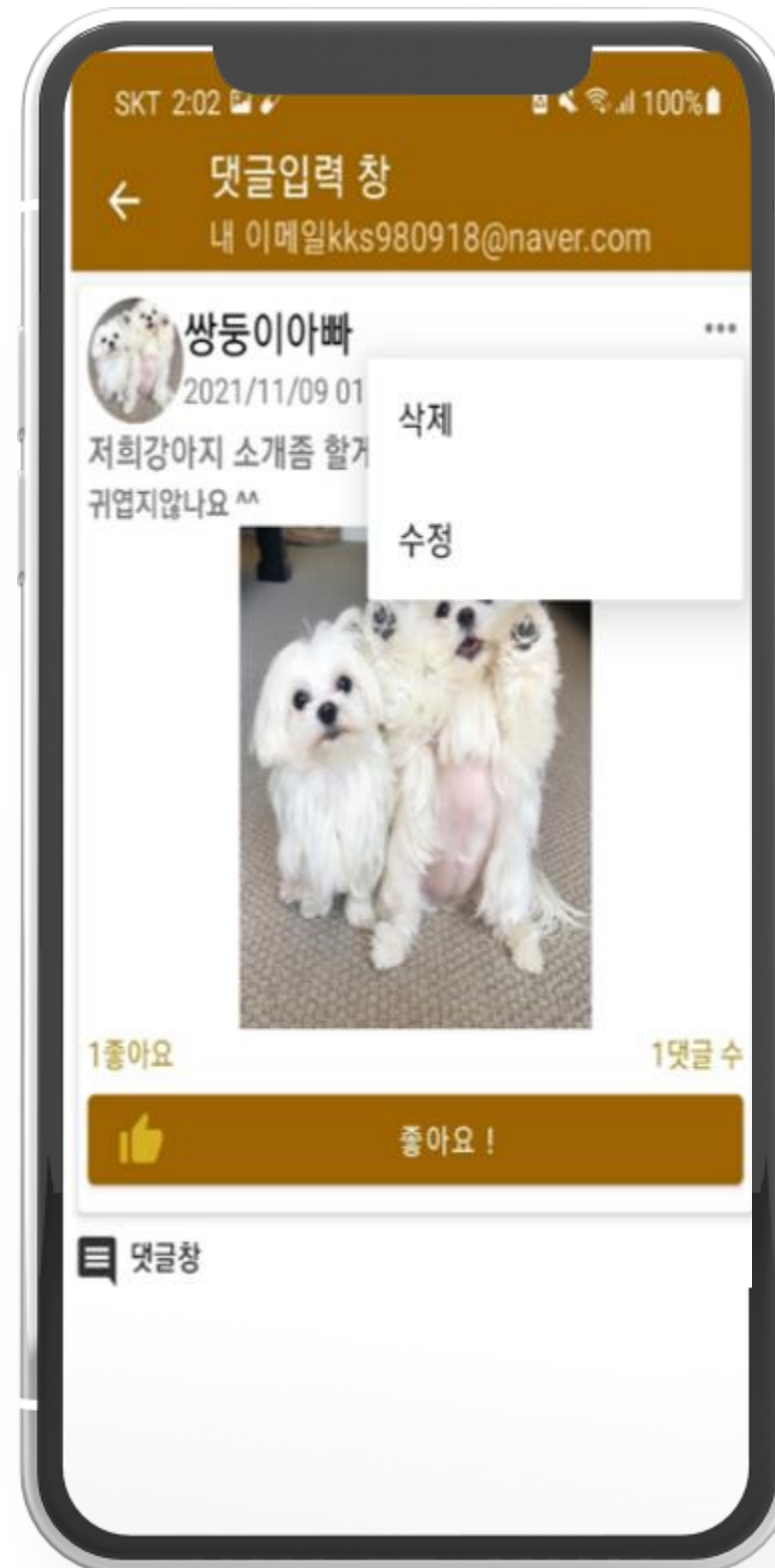
- 제목과 사진을 올리는 칸과 내용 3가지로 구성되어있다.
- 사진은 갤러리에 접근 할 수 있는 권한을 묻는 메시지가 나오고 그 권한을 수락한 후에만 갤러리또는 카메라로 사진을 업로드 할 수 있다.
메세지 창은 Dialog 기능을 이용해 직접 안내문구를 만들었다.

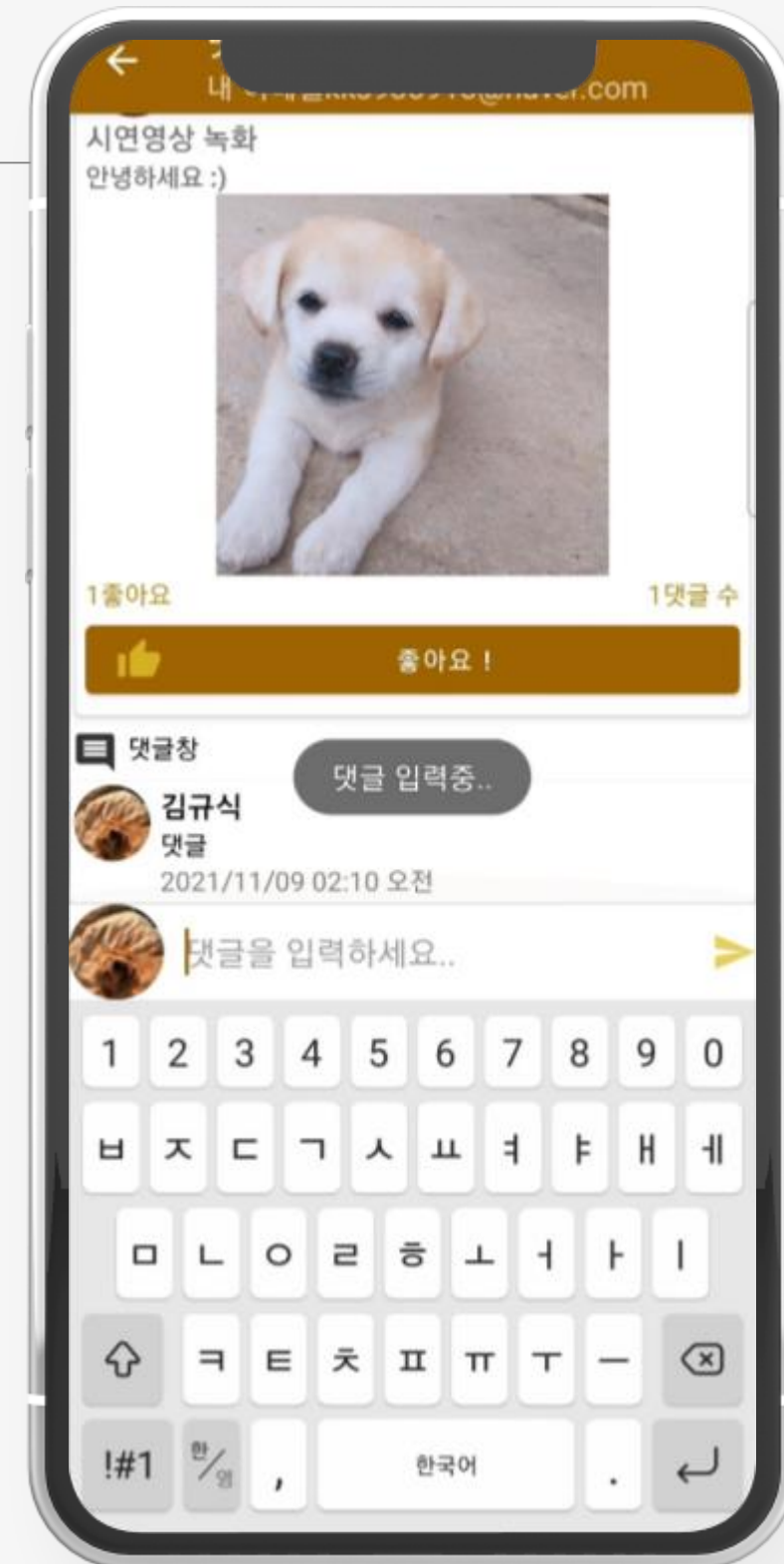
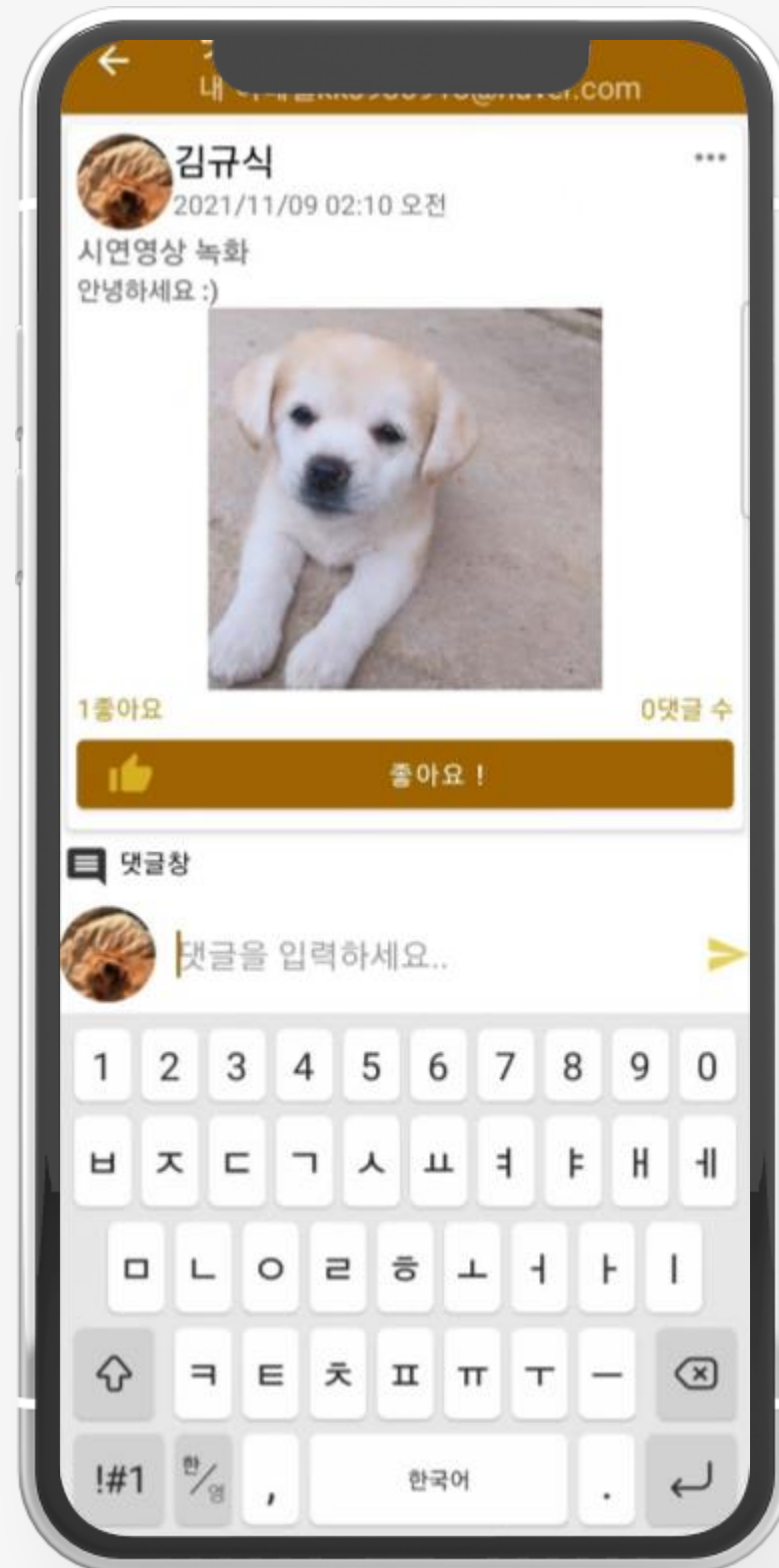
디자인

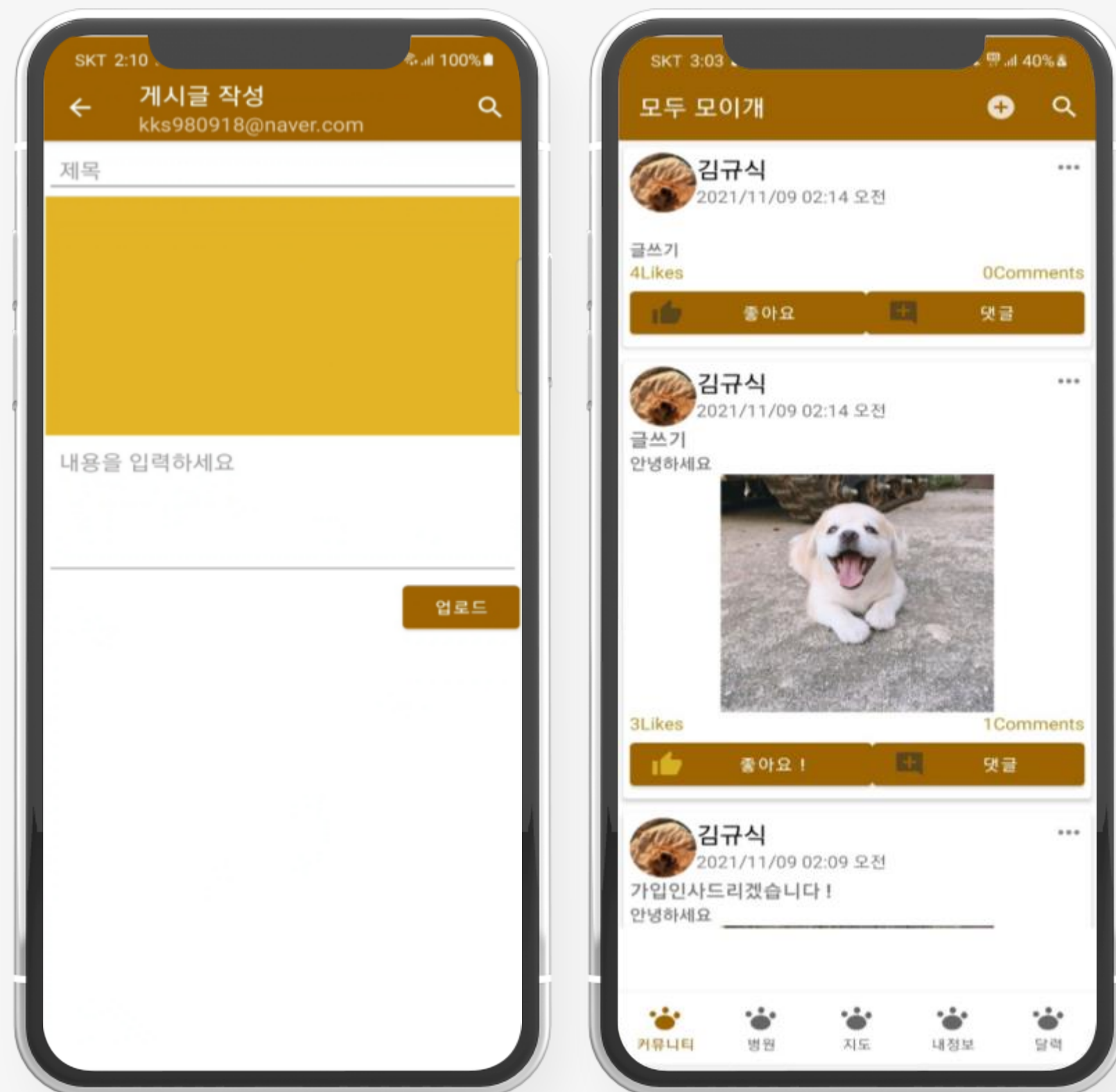
- Adobe Illustrator, Photoshop, Sketch











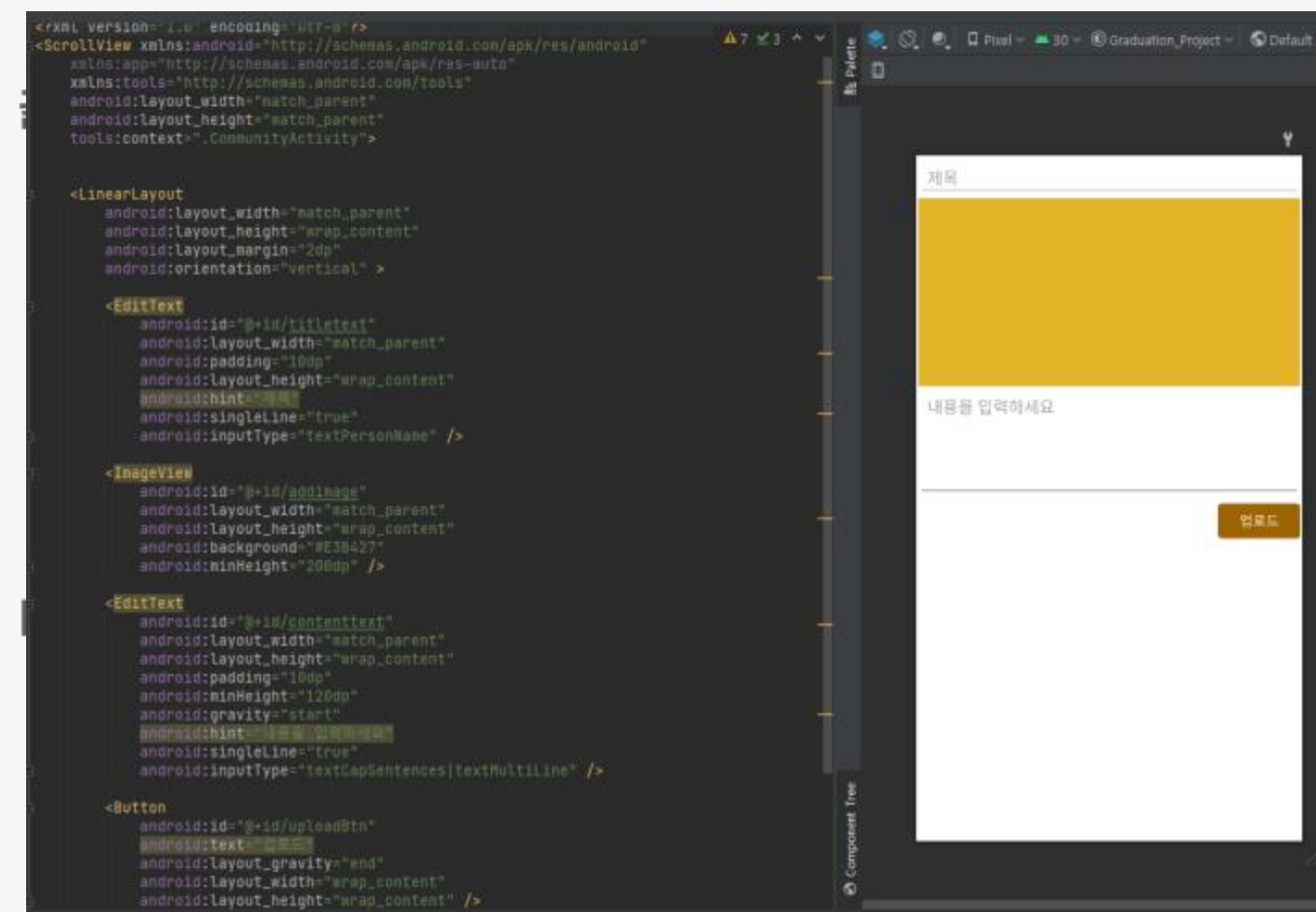
기능 설명

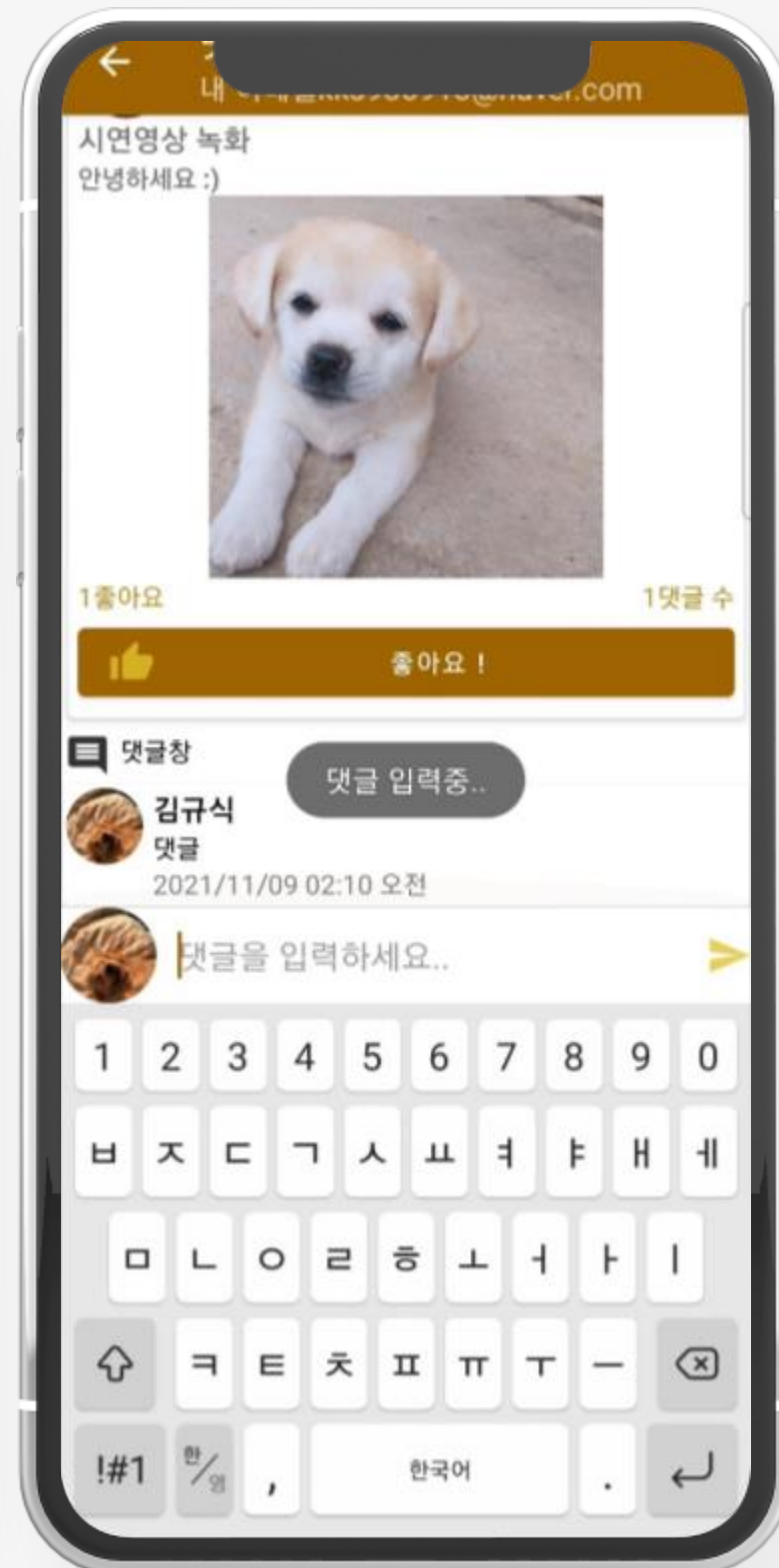
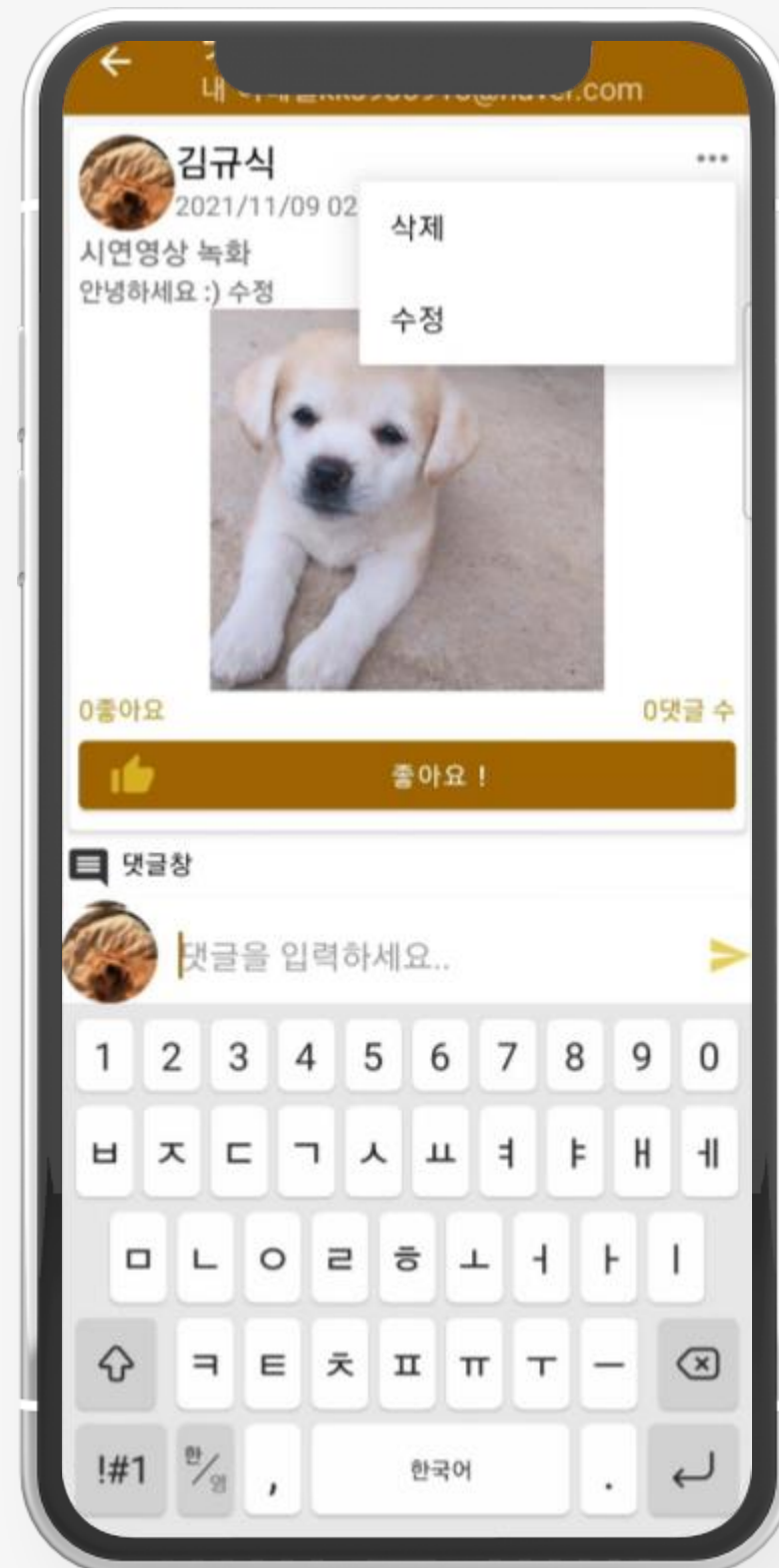
```

databaseReference = FirebaseDatabase.getInstance().getReference();
//카메라 접근
private static final int CAMERA_REQUEST_CODE = 100;
private static final int STORAGE_REQUEST_CODE = 200;
private static final int IMAGE_PICK_CAMERA_CODE = 300;
private static final int IMAGE_PICK_GALLERY_CODE = 400;
    
```

```

//허락
String[] cameraPermissions;
String[] storagePermissions;
    
```





```
public class AdapterComments extends RecyclerView.Adapter<AdapterComments.MyHolder> {
    Context context;
    List<ModelComment> commentList;
    String myUid, postId;

    public AdapterComments(Context context, List<ModelComment> commentList, String myUid, String postId) {
        this.context = context;
        this.commentList = commentList;
        this.myUid = myUid;
        this.postId = postId;
    }

    @NonNull
    @Override
    public MyHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
        View view = LayoutInflater.from(context).inflate(R.layout.row_comments, viewGroup, attachToRoot: false);
        return new MyHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull MyHolder myholder, int i) {
        String uid = commentList.get(i).getUid();
        String name = commentList.get(i).getName();
        String email = commentList.get(i).getEmail();
        String image = commentList.get(i).getUp();
        String cid = commentList.get(i).getCid();
        String comment = commentList.get(i).getComment();
        String timestamp = commentList.get(i).getTimestamp();

        Calendar calendar = Calendar.getInstance(Locale.getDefault());
        try {
            calendar.setTimeInMillis(Long.parseLong(timestamp));
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        String pTime = DateFormat.format("yyyy/MM/dd hh:mm aa", calendar).toString();

        myholder.nameC.setText(name);
        myholder.commentC.setText(comment);
        myholder.timeC.setText(pTime);

        return;
    }

    String timeStamp = String.valueOf(System.currentTimeMillis());
    DatabaseReference ref = FirebaseDatabase.getInstance().getReference().child("Posts").child(postId).child("Comments");

    HashMap<String, Object> hashMap = new HashMap<>();

    hashMap.put("cid", timeStamp);
    hashMap.put("comment", comment);
    hashMap.put("timestamp", timeStamp);
    hashMap.put("uid", myUid);
    hashMap.put("uEmail", myEmail);
    hashMap.put("uOp", myOp);
    hashMap.put("uName", myName);

    ref.child(timeStamp).setValue(hashMap)
        .addOnSuccessListener(new OnSuccessListener<Void>() {
            @Override
            public void onSuccess(Void unused) {
                pd.dismiss();
                Toast.makeText(context, CommunityDetailActivity.this, "댓글 입력중..", Toast.LENGTH_SHORT).show();
                commentEt.setText("");
                updateCommentCount();
            }
        })
        .addOnFailureListener(new OnFailureListener() {
            @Override
            public void onFailure(@NonNull Exception e) {
                pd.dismiss();
                Toast.makeText(context, CommunityDetailActivity.this, ""+e.getMessage(), Toast.LENGTH_SHORT).show();
            }
        });
}

private void updateCommentCount() {
    mProcessComment = true;
    DatabaseReference ref = FirebaseDatabase.getInstance().getReference().child("Posts").child(postId);
    ref.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if (mProcessComment) {
                String comments = "" + dataSnapshot.child("pComments").getValue();
                int newCommentVal = Integer.parseInt(comments) + 1;
                ref.child("pComments").setValue("" + newCommentVal);
            }
        }
    });
}
```

핵심 코드

```
    },
    myHolder.likeBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            //게시글에 좋아요 총 갯수 , 누가 버튼 눌렀는지
            final int pLikes = Integer.parseInt(postList.get(i).getpLikes());
            mProcessLike = true;

            final String postId = postList.get(i).getPostId();
            likeRef.addValueEventListener(new ValueEventListener() {
                @Override
                public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                    if (mProcessLike) {
                        if (dataSnapshot.child(postId).hasChild(myUid)) {
                            //좋아요가 눌러져있고 지울때
                            postsRef.child(postId).child("pLikes").setValue("" + (pLikes-1));
                            likeRef.child(postId).child(myUid).removeValue();
                            mProcessLike = false;
                        }
                        else {
                            //좋아요 안눌름
                            postsRef.child(postId).child("pLikes").setValue("" + (pLikes+1));
                            likeRef.child(postId).child(myUid).setValue("Liked"); //set any value
                            mProcessLike = false;
                        }
                    }
                }
            })
        }
    })
}
```

```
private void setLikes() {
    final DatabaseReference likeRef = FirebaseDatabase.getInstance().getReference().child("Likes");
    likeRef.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if (dataSnapshot.child(postId).hasChild(myUid)) {
                //유저가 좋아요 눌렀면
                //아이콘 색과 버튼이 바뀌고 / 문구가 바뀜
                likeBtn.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic_liked, top: 0, right: 0, bottom: 0);
                likeBtn.setText("좋아요 !");
            }
            else {
                //누르지 않으면
                likeBtn.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic_like_black, top: 0, right: 0, bottom: 0);
                likeBtn.setText("좋아요");
            }
        }

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

private void likePost() {
    mProcessLike = true;

    final DatabaseReference likeRef = FirebaseDatabase.getInstance().getReference().child("Likes");
    final DatabaseReference postsRef = FirebaseDatabase.getInstance().getReference().child("Posts");

    likeRef.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            if (mProcessLike) {
                if (dataSnapshot.child(postId).hasChild(myUid)) {
                    //좋아요가 눌러져있고 지울때
                    postsRef.child(postId).child("pLikes").setValue("" + (Integer.parseInt(pLikes)-1));
                    likeRef.child(postId).child(myUid).removeValue();
                    mProcessLike = false;
                }
                else {
                    //좋아요 안눌름
                    postsRef.child(postId).child("pLikes").setValue("" + (Integer.parseInt(pLikes)+1));
                    likeRef.child(postId).child(myUid).setValue("Liked"); //set any value
                    mProcessLike = false;
                }
            }
        }
    });
}
```

감사합니다!

20181763 김규식

발표마치겠습니다

2021 1조 졸업작품 포트폴리오

김규식

kuchic0918@gmail.com

010-4072-9809

