같공

< 같이 공부하는 작은 공간 >

20180502 안시현
20171703 정태원
20182082 박주현
20182274 박은지

같공 소개

앱 타겟층 : 대학생

-같은 대학교 학생들의 정보 교류

-팀프로젝트 참여기능 제공

-대면/비대면 스터디 제공

-

같공 주요 기능

01

채팅 기능

팀프로젝트나 스터디방을 USER가 개설하면 자동으로 채팅방이 형성되고 소속유저들끼리 채팅을 할 수 있습니다.

02

프로젝트 참여 기능

프로젝트 참여자를 모집하기 위해 글을 작성하고 해당 게시물을 사용자가 읽도록 합니다.

03

ZOOM 참여 기능

프로젝트에 ZOOM SDK를 연동하여 비대면 스터디시에 편리함을 제공합니다.

자동화 가능 부분

-Study 방에 게시글을 작성시 자동으로 Zoom 방을 개설합니다.

-개설된 방에 접속해서 참여하기를 선택하면 해당 방으로 이동합니다.

-Project 방에 게시글 작성시 자동으로 채팅방이 개설됩니다.

-

경쟁앱과 비교

ZOOM 참여

-스터디 기능을 제공하는 가장 대표적인 앱 열품타는 비대면 스터디를 할 때에 각자 공부를 한 시간을 비교할 수 있는 기능이 중점적으로 되어있습니다. 같공은 ZOOM을 연동하여 비대면 스터디도 대면처럼 서로 공부하는 모습을 볼 수 있고 비대면 스터디 중에 채팅도 가능합니다. 또한 대화도 나눌 수 있습니다.

경쟁앱과 비교

채팅 기능

대학 커뮤니티 앱 중 가장 대표적인 에브리타임과 비교하여 에브리타임은 1:1로 쪽지를 나누는 방식으로 채팅을 합니다. 같공은 같은 팀프로젝트에 소속되어 있는 유저들, 같은 스터디에 소속되어 있는 유저들끼리 단체로 채팅방을 이용해 채팅이 가능합니다. 따라서 소속감을 느끼고 장기적으로 앱을 이용하게 될 가능성이 높습니다.

개선점과 확장성

개선점:
(중간점검)

개선점으로는 파이어베이스를 구현하여 데이터 저장하는데에 있어 많은 제약이 사라졌다. Preference를 활용해 데이터를 넘겨주었던 기존의 방식에 비해 데이터를 체계적으로 저장할 수 있다.

확장성:

과목코드를 도입하여 과목에 따라 분류된 게시글을 제공한다면 자신의 과목과 관련된 정보를 더욱 빠르고 쉽게 공유할 수 있다.
과목과 관련된 프로젝트방과 질문방에 학생들이 방문하여 강의실에서 벗어난 활용을 통해 수업을 통해 배운 지식을 체득할 수 있다.

layout

Constraintlayout을 통해 edit text와 button의 위치를 설정

Edit text

이메일과 비밀번호를 Edit text를 통해 구현하고 hint를 통해 입력해야 할 값을 지정

로그인

Button

로그인 버튼과 회원가입 버튼 클릭 시에 로그인은 Home으로, 회원가입 버튼 클릭 시에는 회원가입 Activity로 이동

Firebase 연동

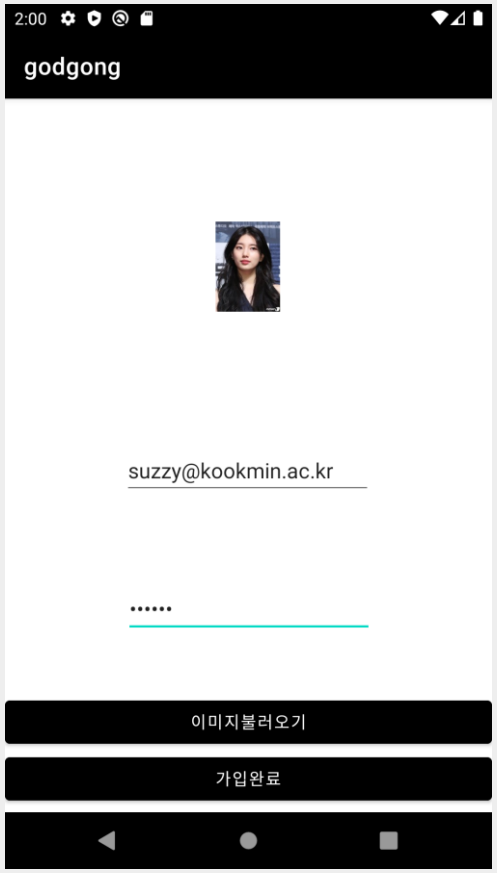
Firebase를 통해 UserAccount 정보를 연동시켜 사용자 정보를 저장하고 불러오는 기능을 수행시킴

layout

Constraintlayout을 통해 edit text와 button, ImageView의 위치를 설정

Edit text

이메일과 비밀번호를 Edit text를 통해 구현하고 hint를 통해 입력해야 할 값을 지정



회원가입

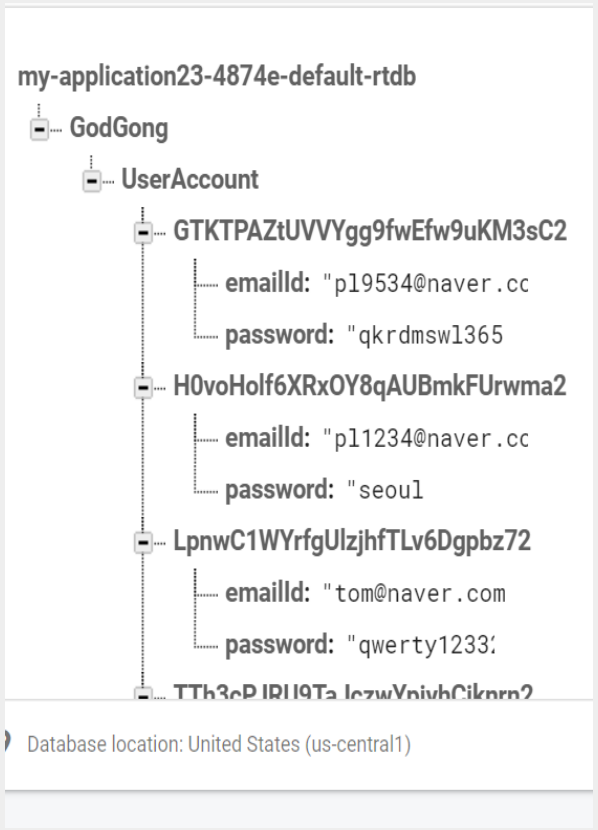
Button

이미지 불러오기 버튼과 가입완료 버튼 클릭 시에 이미지 버튼은 이미지를 불러오는 기능을 수행하고 가입완료 버튼을 클릭 시에는 다시 로그인 화면으로 이동

Firebase 연동

Firebase를 통해 UserAccount 정보를 연동시켜 사용자 정보를 저장하고 불러오는 기능을 수행시키며, 프로필 사진을 지정


10:02
이메일을 입력하세요
비밀번호를 입력하세요
로그인
회원가입



Firebase + Android

회원가입 버튼을 클릭 시에 회원가입 정보를 입력하게 함

회원가입

- 파이어베이스 접근 권한 갖기

FirebaseAuth firebaseAuth = FirebaseAuth.getInstance();

- 가입 버튼 : 신규 계정이 friebase의 realtime에 등록된다.

로그인

- Authentication에 등록만 되어있다면 로그인에 성공하게 된다.

```java
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_login);

    mFirebaseAuth = FirebaseAuth.getInstance();
    mDatabaseRef = FirebaseDatabase.getInstance().getReference( path: "GodGong");

    mEtEmail = findViewById(R.id.et_email);
    mEtPwd = findViewById(R.id.et_pwd);


    Button btn_login = findViewById(R.id.btn_login);
    btn_login.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {

            //로그인 요청
            String strEmail = mEtEmail.getText().toString();
            String strPwd = mEtPwd.getText().toString();

            mFirebaseAuth.signInWithEmailAndPassword(strEmail, strPwd).addOnCompleteListener( activity: LoginActivity.this,
                @Override
                public void onComplete(@NonNull Task<AuthResult> task) {
                    if(task.isSuccessful()) {
                        //로그인 성공!!!
                        Intent intent = new Intent(LoginActivity.this, MainActivity1.class);
                        startActivity(intent);
                        finish(); //현재 액티비티 파괴

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

```java
mEtEmail = findViewById(R.id.et_email);
mEtPwd = findViewById(R.id.et_pwd);
mBtnRegister = findViewById(R.id.btn_register);
mBtnRegister.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        //회원가입 처리 시작
        String strEmail = mEtEmail.getText().toString();
        final String strPwd = mEtPwd.getText().toString();

        // Firebase Auth 진행
        mFirebaseAuth.createUserWithEmailAndPassword(strEmail, strPwd).addOnCompleteListener( activity: RegisterActivity.this, new OnComp
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if (task.isSuccessful()) {
                    FirebaseUser firebaseUser = mFirebaseAuth.getCurrentUser();

                    UserAccount account = new UserAccount();
                    account.setIdToken(firebaseUser.getUid());
                    account.setEmailId(firebaseUser.getEmail());
                    account.setPassword(strPwd);
                     account.setProfile(imageView);
```

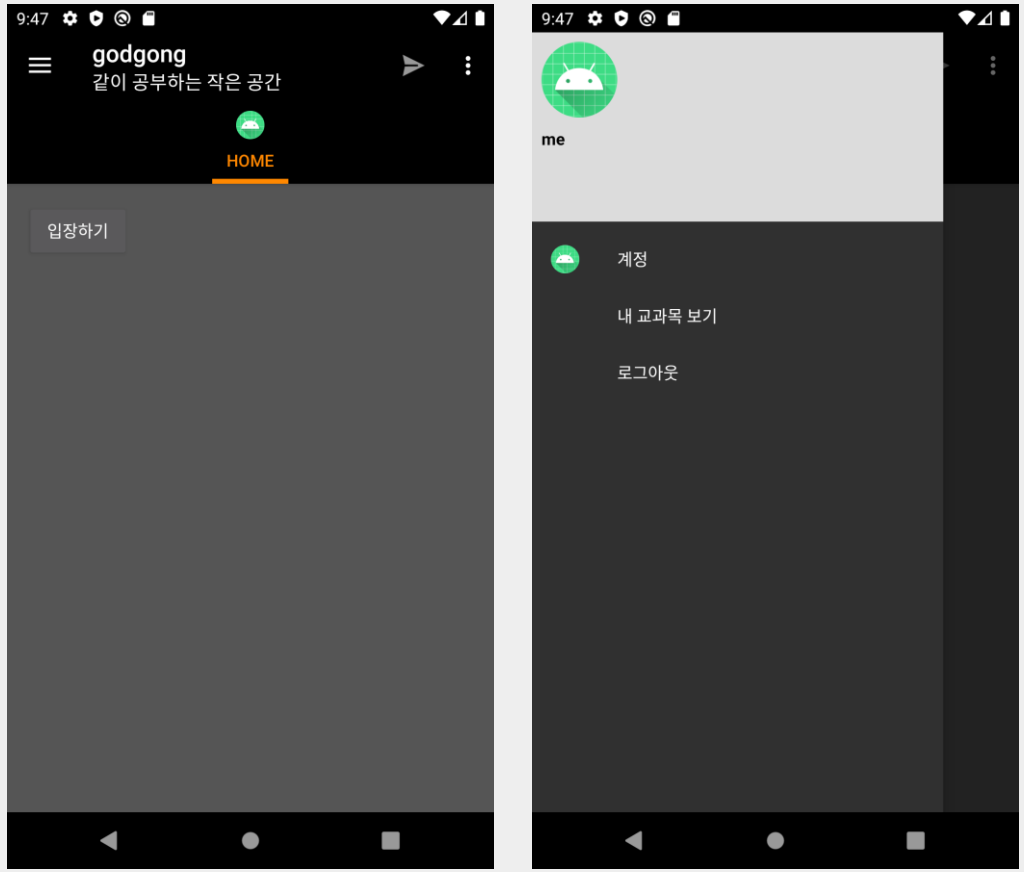
Firebase를 연동한 로그인 및 회원가입 구현화면

DrawerLayout

메뉴를 왼쪽에서 오른쪽으로 열기 위해 필요한 Layout, DrawerLayout을 최상단에 배치해야 하며, 크게 두 가지 부분으로 구성 된다.

Header

HeaderLayout 속성의 속성 값 지정



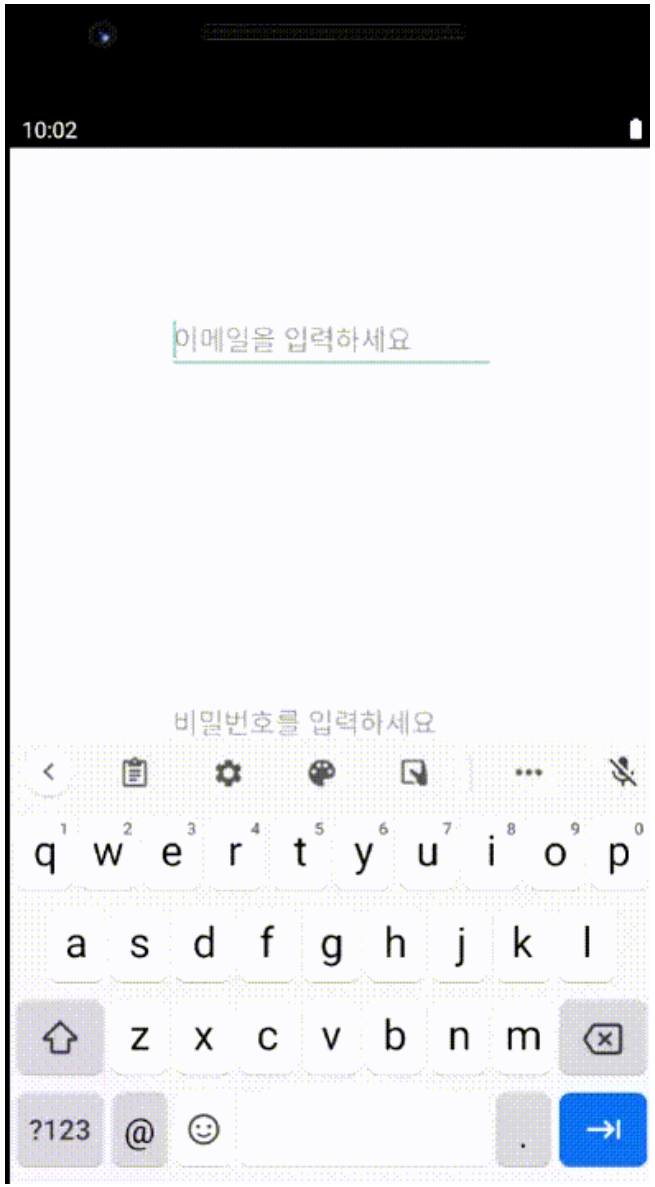
Navigation Drawer

Content 영역

Menu 레이아웃 리소스가 정의된 파일의 리소스 ID를 menu 속성 값으로 지정

Navigation View

DrawerLayout 안쪽에 Toolbar와 NavigationView를 생성


10:02
이메일을 입력하세요
비밀번호를 입력하세요
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123
@
.

```
<androidx.drawerlayout.widget.DrawerLayout xmlns:android="http://schemas.android.com/a
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/layout_drawer"
    tools:context=".MainActivity1">

    <!--    1. 화면을 구성하는 Content 영역-->
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical" >

        <com.google.android.material.appbar.AppBarLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:theme="@style/AppTheme.Appbar">

            <androidx.appcompat.widget.Toolbar
                android:id="@+id/toolbar"
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize">
            </androidx.appcompat.widget.Toolbar>

            <com.google.android.material.tabs.TabLayout
                android:id="@+id/layout_tab"
```

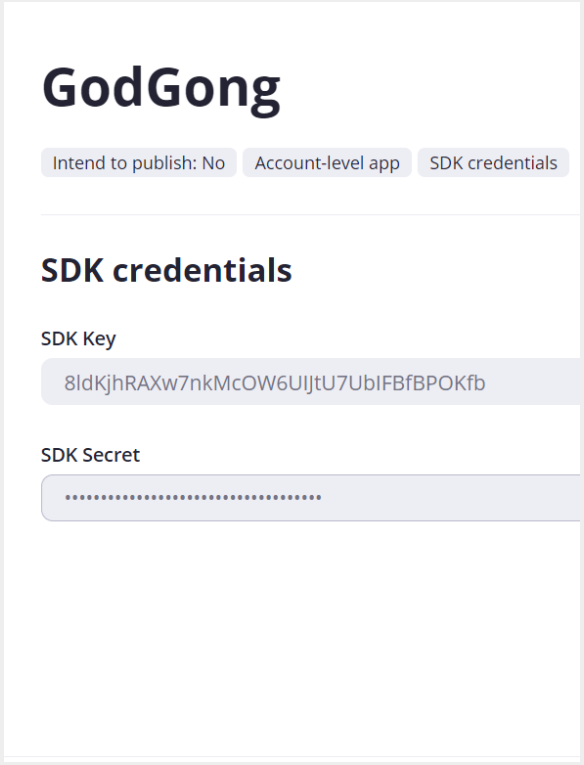
```
            </androidx.appcompat.widget.Toolbar>

            <com.google.android.material.tabs.TabLayout
                android:id="@+id/layout_tab"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                app:tabTextColor="@color/white"
                app:tabSelectedTextColor="#FF8800"
                app:tabIndicatorColor="#FF8800"
                app:tabIndicatorHeight="4dp"
                app:tabMode="fixed"
                app:tabGravity="fill">
            </com.google.android.material.tabs.TabLayout>

        </com.google.android.material.appbar.AppBarLayout>

        <androidx.viewpager.widget.ViewPager
            android:id="@+id/pager"
            android:layout_width="match_parent"
            android:layout_height="match_parent"/>

    </LinearLayout>

    <!--    2. 왼쪽 사이드 메뉴-->
    <com.google.android.material.navigation.NavigationView
        android:id="@+id/nav"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_gravity="start"
        app:headerLayout="@layout/activity_sidebar"
        app:menu="@menu/drawerlayout"/>
```

Drawer Navigation layout resource

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:id="@+id/menu_account"
        android:title="계정"
        android:icon="@mipmap/ic_launcher"/>

    <item android:id="@+id/menu_look"
        android:title="내 교과목 보기"/>

    <item android:id="@+id/menu_logout"
        android:title="로그아웃"/>

</menu>
```

```java
//네비게이션뷰에 아이템선택 리스너 추가
navigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {
    @Override
    public boolean onNavigationItemSelected(@NonNull MenuItem item) {
        item.setChecked(true);

        String getId = firebaseUser.getEmail();
        int id = item.getItemId();
        String title = item.getTitle().toString();

        Context context = null;

        if(id == R.id.menu_account) {
            Toast.makeText(context, text: title + "회원님의 아이디는" + getId , Toast.LENGTH_SHORT).show();
        }
        else if (id==R.id.menu_look) {
            Toast.makeText(context, text: title + ":내 교과목을 확인합니다.", Toast.LENGTH_SHORT).show();
        }
        else if (id == R.id.menu_logout) {
            Intent i = new Intent(getApplicationContext(),LoginActivity.class);
            startActivity(i);
        }
        drawerLayout.closeDrawer(GravityCompat.START);
        return false;
    }
```

Navigation의 Item 클릭에 대한 이벤트 처리



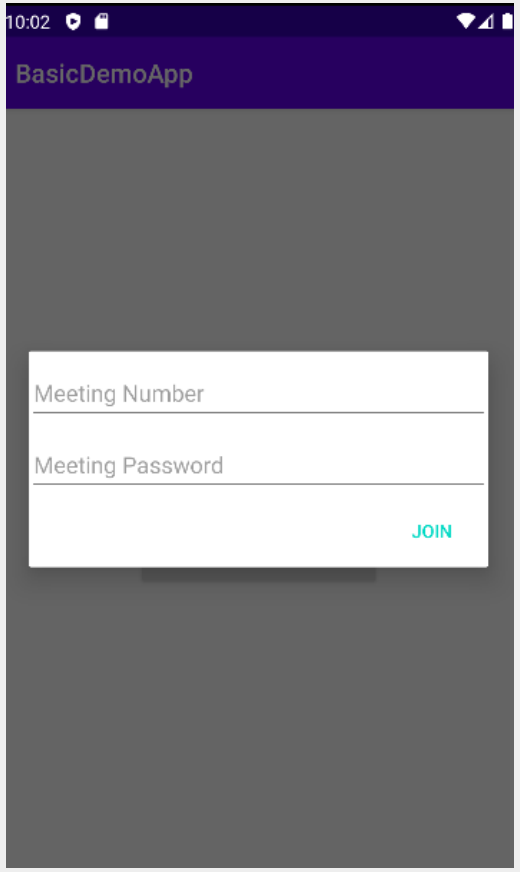
Zoom sdk + Android

Zoom 계정 개설을 통해 Zoom SDK android용 API 키를 생성, 애플리케이션 이름을 지정하고 생성된 SDK key를 확인,

프로젝트에 Zoom SDK를 추가, 프로젝트에 종속성을 추가

Zoom SDK 구현을 위해 인터넷 권한 추가, Zoom SDK Key추가, Zoom 초기화를 진행

회의를 만들어서 Meeting number와 meeting password 입력을 통해 회의에 참여




```
public void initializeSdk(Context context) {
    ZoomSDK sdk = ZoomSDK.getInstance();
    // TODO: Do not use hard-coded values for your key/secret in your app in production!
    ZoomSDKInitParams params = new ZoomSDKInitParams();
    params.appKey = "8ldKjhRAXw7nkMcOW6UIJtU7UbIFBfBPOKfb"; // TODO: Retrieve your SDK key and enter it here
    params.appSecret = "nLu6cbw9EeJWNtH2g2RU2ftziQXVQ4W4jOet"; // TODO: Retrieve your SDK secret and enter it here
    params.domain = "zoom.us";
    params.enableLog = true;
    // TODO: Add functionality to this listener (e.g. logs for debugging)
    ZoomSDKInitializeListener listener = new ZoomSDKInitializeListener() {
        /**
         * @param errorCode {@link us.zoom.sdk.ZoomError#ZOOM_ERROR_SUCCESS} if the SDK has been initialized successfully.
         */
        @Override
        public void onZoomSDKInitializeResult(int errorCode, int internalErrorCode) { }

        @Override
        public void onZoomAuthIdentityExpired() { }
    };
    sdk.initialize(context, listener, params);
}
```

Zoom Sdk 연동

질문 작성

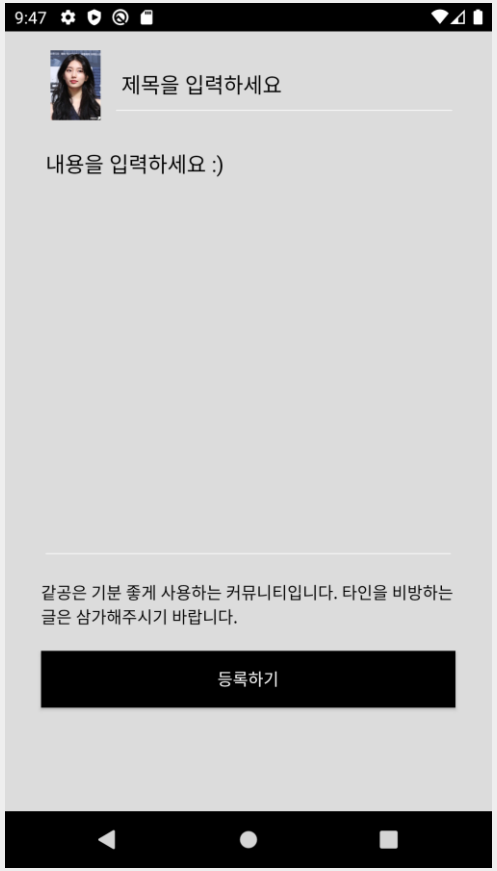
작성한 질문을 구조화된 데이터로 저장

게시물 작성

Firebase realtime data에 post.class에 글을 담아 작성합니다.

작성자 사진

Firebase storage에서 작성자의 사진을 가져옵니다.



날짜

글을 작성하는 시간을 post로 넘깁니다.

getkey, push

Getkey함수를 통해 작성할 post글의 키를 부여받습니다.


10:02
11-29
dkstlgus7571@naver.com
sdf
11-29
dog@dog.com
writing
11-29
dkstlgus7571@naver.com
I have a question
11-29
dog@dog.com
write titleslek
질문하기

질문 작성

```
FirebaseStorage storage = FirebaseStorage.getInstance();
StorageReference storageReference = storage.getReference();
StorageReference pathReference = storageReference.child("images");
if(pathReference == null){
    Toast.makeText( context: DetailActivity.this,  text: "저장소에 사진이 없습니다.", Toast.LENGTH_SHORT).show();}
else{
    Toast.makeText( context: DetailActivity.this,  text: "저장소에 사진이 있습니다.", Toast.LENGTH_SHORT).show();


    StorageReference submitProfile = storageReference.child("images/"+email);
```

Firebase의 StorageReference를 활용하여 저장된 사용자의 프로필 사진과 정보를 가져와 출력합니다

-

질문 작성

Post.class를 작성하여 Firebase realtime data에 등록할 질문과 관련된 정보를 구조화하여 저장합니다

```
Post post = new Post();

post.setEmailId(firebaseUser.getEmail());
post.setTitle_et(strTitle);
post.setContent_et(strContent);
post.setDate(getTime);
post.setWriterId(firebaseUser.getUid());
String key = mDatabaseRef.child("projectposts").push().getKey();
post.setToken(key);
```

질문 작성

작성한 Post.class

```
class Post {

    public String title_et;
    public String content_et;
    public int starCount = 0;
    public ImageView image;
    private String emailId;   //이메일 아이디
    private String token;
    private String Date;
    private String writerId;

    public Post(String userid, String title_et, String content_et) {

        this.title_et = title_et;
        this.content_et = content_et;

    }
```

질문 작성

Post.class형식으로 저장된 데이터

채팅방

채팅 기능

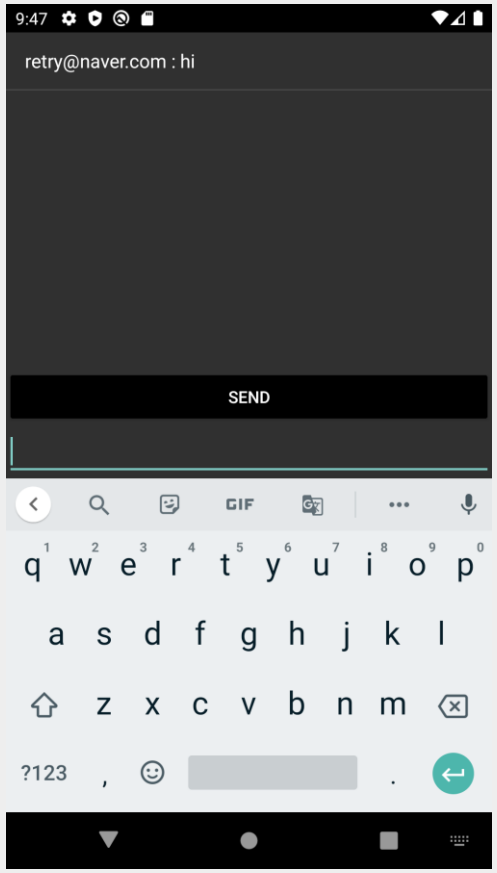
팀프로젝트 채널, 스터디방 채널에서 user가 참가한 방에서 어디에서든지 채팅기능을 이용할 수 있습니다. 채팅하기 버튼을 누르면 채팅이 가능합니다.

Firebase에 메시지를 저장

Firebase에 메시지가 저장되어 앱이 종료되었다가 켜져도 메시지가 그대로 저장되어있습니다.

채팅 자동화

.User가 글을 작성하면 작성자가 작성한 제목으로 파이어베이스에 자동으로 채팅방이 생성된다.

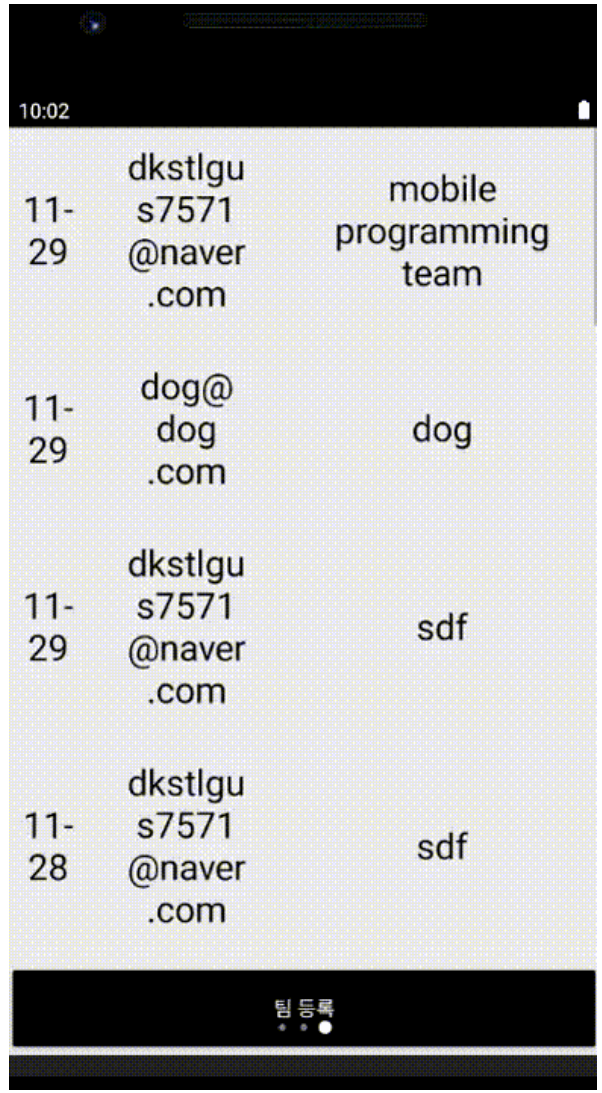


메시지 입력

메시지를 작성하고 Send 버튼을 누르면 메시지가 전송됩니다.

Listview 이용

메시지가 원활하게 올라가게 하기 위해 listvew를 이용하였습니다.


10:02
11-29
dkstlgus7571@naver.com
mobile programming team
11-29
dog@dog.com
dog
11-29
dkstlgus7571@naver.com
sdf
11-28
dkstlgus7571@naver.com
sdf
팀 등록

채팅 기능 구현

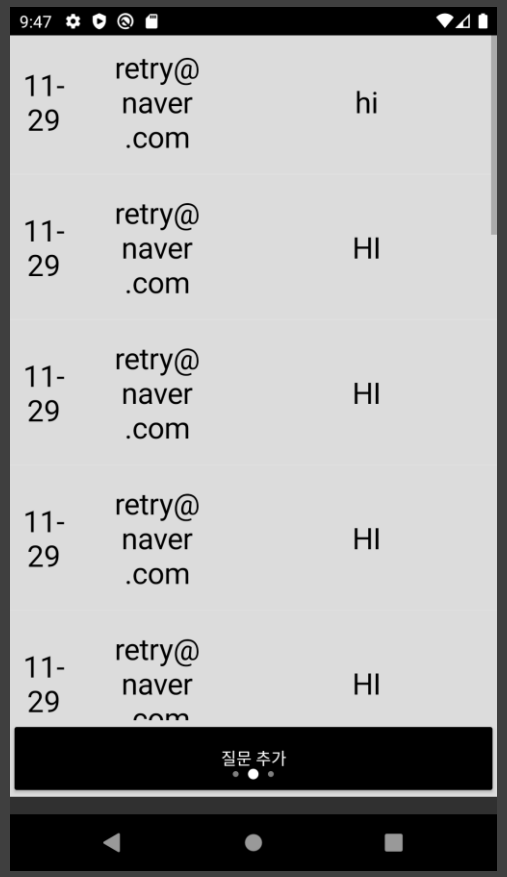
```
private void openChat(String chatName) {
    // 리스트 어댑터 생성 및 세팅
    final ArrayAdapter<String> adapter

            = new ArrayAdapter<~>( context: this, android.R.layout.simple_list_item_1, android.R.id.text1);
    chat_view.setAdapter(adapter);

    // 데이터 받아오기 및 어댑터 데이터 추가 및 삭제 등..리스너 관리
    databaseReference.child("chat").child(chatName).addChildEventListener(new ChildEventListener() {
        @Override
        public void onChildAdded(DataSnapshot dataSnapshot, String s) {
            addMessage(dataSnapshot, adapter);
            Log.e( tag: "LOG",  msg: "s:"+s);
        }

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

        }

        @Override
        public void onChildRemoved(DataSnapshot dataSnapshot) {
            removeMessage(dataSnapshot, adapter);
        }

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

        }

        @Override
        public void onCancelled(DatabaseError databaseError) {

        }
    });
}
```

OnChildAdded(): 항목 목록을 검색하거나 항목 목록에 대한 추가를 수신 대기합니다.

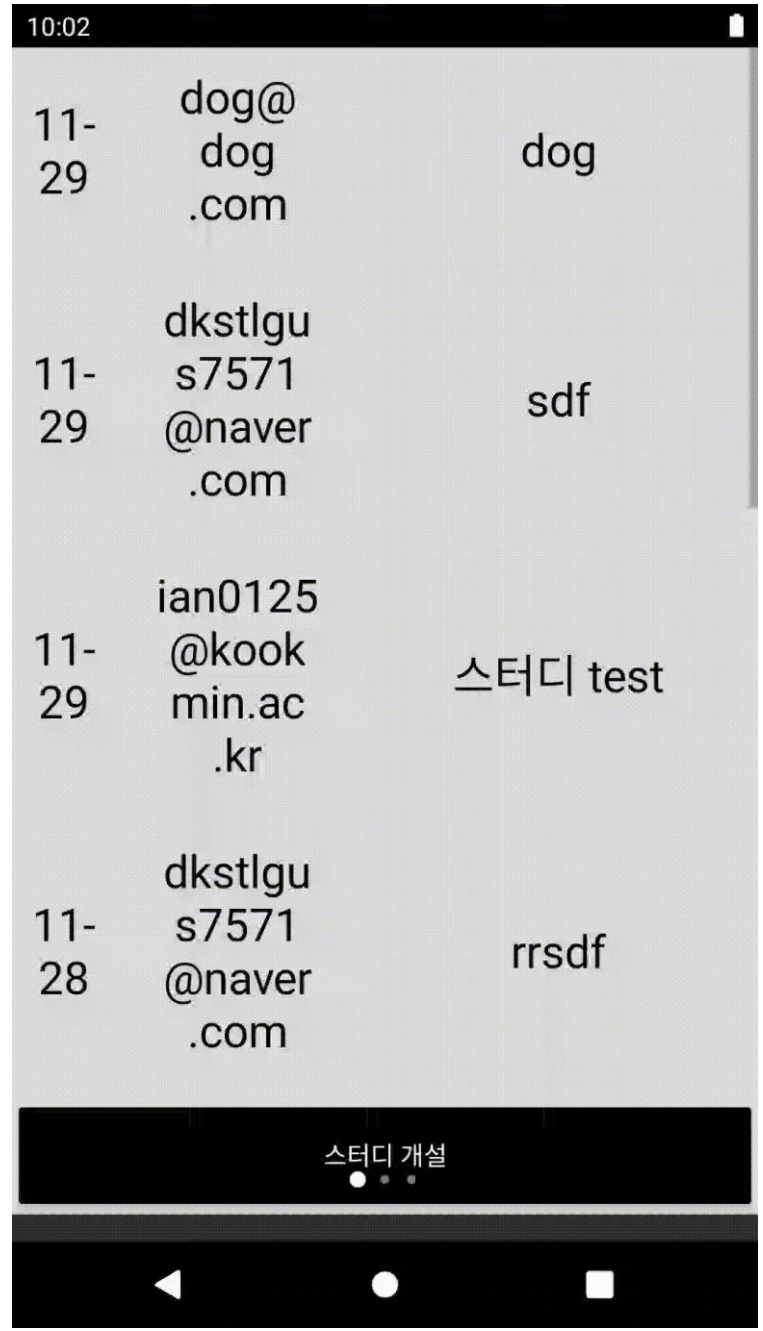
OnChildChanged():목록의 항목에 대한 변경을 수신 대기합니다.

OnChildRemoved():목록의 항목 삭제를 수신 대기합니다.

OnChildMoved():순서 있는 목록의 항목 순서 변경을 수신 대기합니다.



ViewPager2를 사용하여 3가지 채널 생성


10:02
11-29
dog@dog.com
dog
11-29
dkstlgus7571@naver.com
sdf
11-29
ian0125@kookmin.ac.kr
스터디 test
11-28
dkstlgus7571@naver.com
rrsdf
스터디 개설

회전

```
mPager.registerOnPageChangeCallback(new ViewPager2.OnPageChangeCallback() {
    @Override
    public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels)
        super.onPageScrolled(position, positionOffset, positionOffsetPixels);
        if (positionOffsetPixels == 0) {
            if(position==3) position=0;
            mPager.setCurrentItem(position);
        }
    }

    @Override
    public void onPageSelected(int position) {
        super.onPageSelected(position);
        if(position==5) position=0;
        position%=num_page;
        mIndicator.animatePageSelected(position);
    }
});
```

각 방이 옆으로 스크롤함에 따라 회전하게끔 합니다.

질문방

질문 작성 게시판 DetailActivity

질문

해결하지 못한 문제를 질문방에 작성합니다.

댓글 작성

질문에 대한 댓글을 작성하여 사용자들 사이의 자유로운 지식교환을 할 수 있습니다.

Comment.class

저장할 댓글의 정보 형식에 적합한 comment 형식으로 firebase에 등록합니다.


10:02
.com
2021-11-29
I have a question
hello, how can I do Java well?
댓글 내용을 입력하세요
등록하기

댓글기능 구현

```
FirebaseStorage storage = FirebaseStorage.getInstance();
StorageReference storageReference = storage.getReference();
StorageReference pathReference = storageReference.child("images");
if(pathReference == null){
    Toast.makeText( context: DetailActivity.this,  text: "저장소에 사진이 없습니다.", Toast.LENGTH_SHORT).show();}
else{
    Toast.makeText( context: DetailActivity.this,  text: "저장소에 사진이 있습니다.", Toast.LENGTH_SHORT).show();


    StorageReference submitProfile = storageReference.child("images/"+email);
```

Firebase의 StorageReference를 활용하여 저장된 사용자의 프로필 사진과 정보를 가져와 출력합니다

-

댓글 기능 구현

새로 작성된 댓글은 Firebase realtime data에 저장됩니다.
Firebase의 데이터가 변형되어 onDataChange 를 통해 기존 화면에 나타난 모든 댓글들을 제거하고 저장된 댓글들을 차례대로 화면에 나타냅니다.

```
ValueEventListener commentListener = new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {

        comment_layout.removeAllViews();

        for (DataSnapshot commentSnapshot : snapshot.getChildren()) {

            // custom_comment 를 불러오기 위한 객체
            LayoutInflater layoutInflater = LayoutInflater.from(DetailActivity.this);

            View customView = layoutInflater.inflate(R.layout.custom_comment,  root null);

            Comment com = commentSnapshot.getValue(Comment.class);
```

댓글 기능 구현

게시글 작성과 마찬가지로
Firebase realtime data에 등록할 댓글을 구조화하여 저장하기 위해
comment.class를 작성합니다 .

```
Comment com = commentSnapshot.getValue(Comment.class);


if (com != null) {

    String userid = com.getId();
    String content = com.getComment();
    String crt_dt = com.getDate();

    ((TextView) customView.findViewById(R.id.cmt_userid_tv)).setText(userid);
    ((TextView) customView.findViewById(R.id.cmt_content_tv)).setText(content);
    ((TextView) customView.findViewById(R.id.cmt_date_tv)).setText(crt_dt);


    // 댓글 레이아웃에 custom_comment 의 디자인에 데이터를 담아서 추가
    comment_layout.addView(customView);
}
```

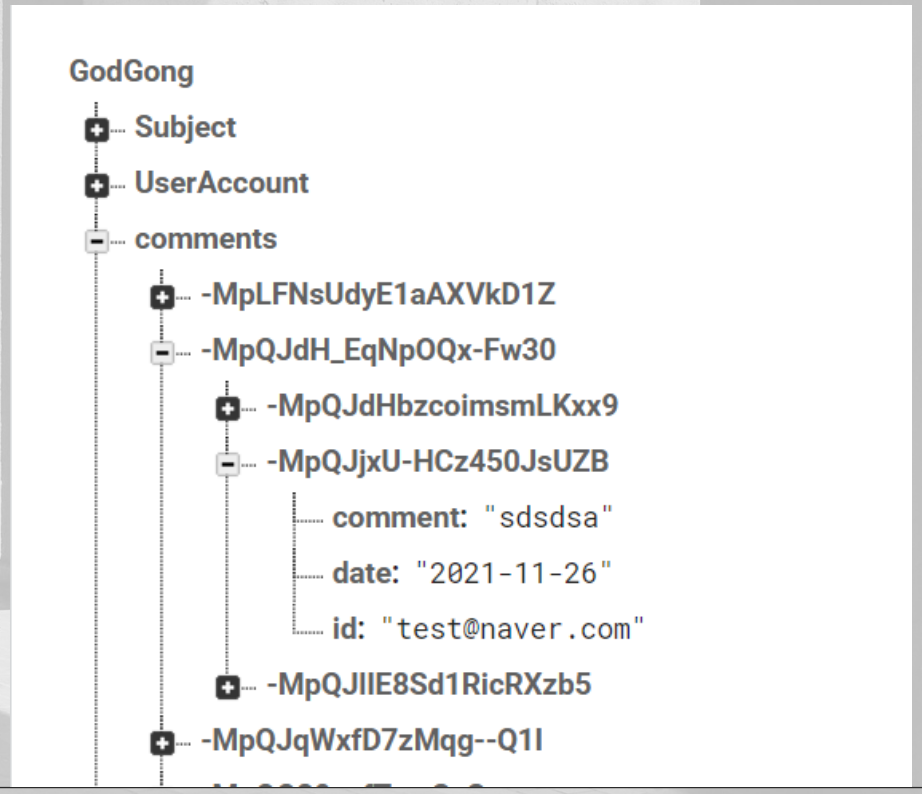
댓글 기능 구현

작성한 comment.class

```java
public class Comment {

    public String comment;
    public String date;
    public String id;
    public Map<String, Boolean> stars = new HashMap<>();

    public Comment(){
        id = "";
        date = "";
        comment="";
    }
```

댓글 기능 구현

comment.class 형식으로 저장된 데이터



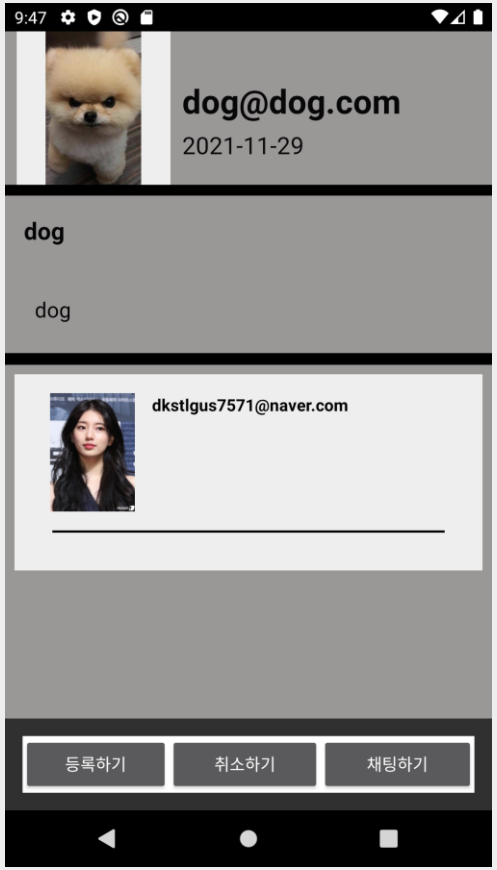
Project참여 게시판

게시글 확인

Project 작성자가 작성한 글을 확인 할 수 있습니다.

참여자 리스트 확인

해당 프로젝트에 참가하고 싶은 인원은 등록하기를 통해 참가의사를 밝힙니다.

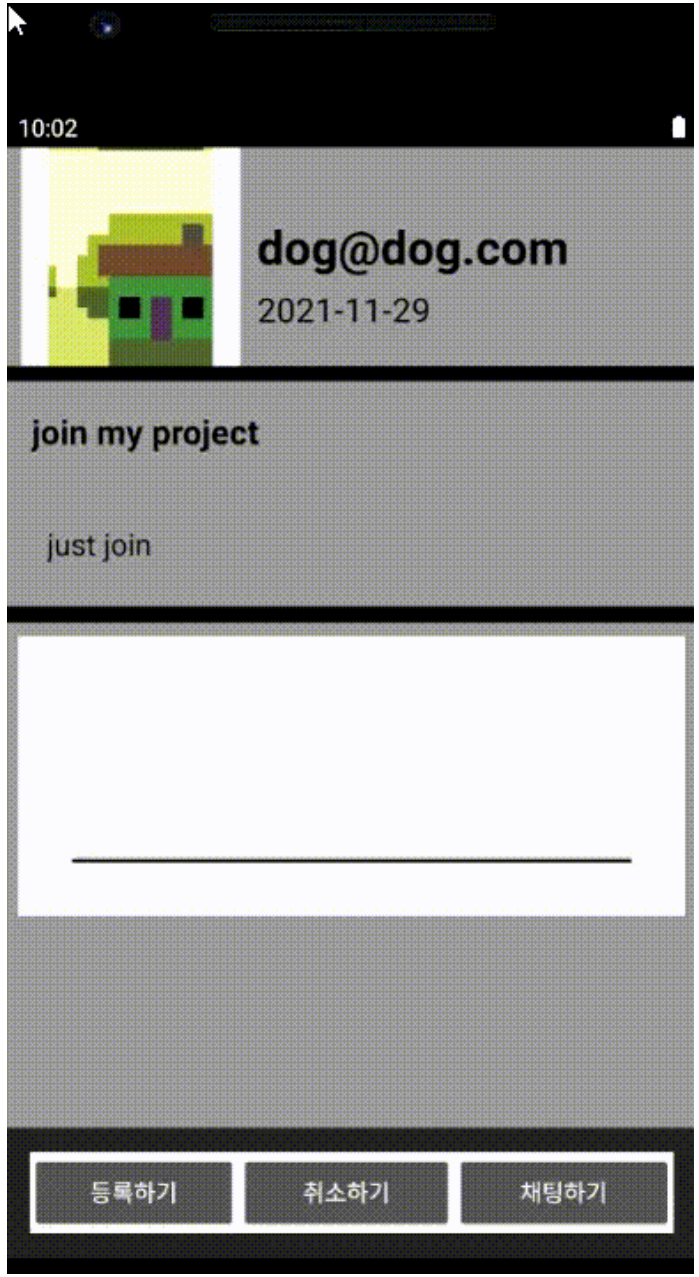


채팅방 기능

List를 클릭하면 해당하는 내용의 게시물의 내용이담긴 인덴스로 넘어가게 됩니다.

작성자 프로필

작성자의 사진을 firebase에서 가져옵니다.

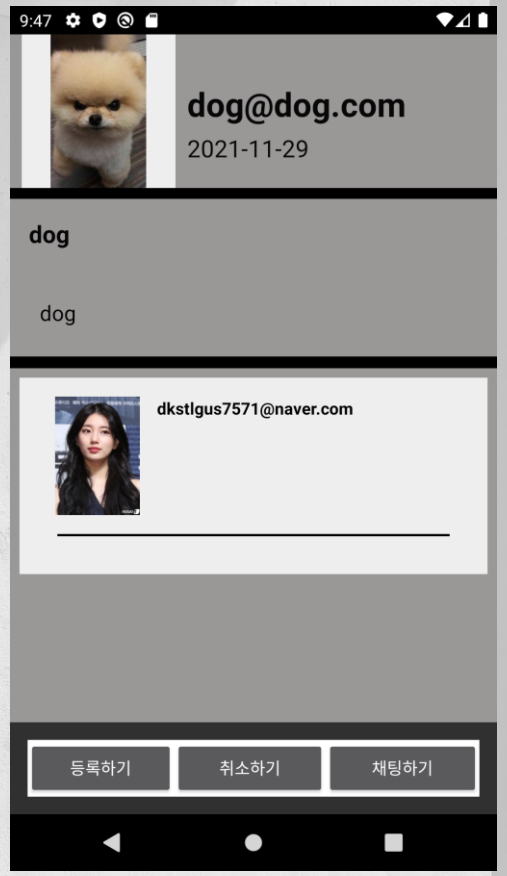

10:02
dog@dog.com
2021-11-29
join my project
just join
등록하기
취소하기
채팅하기

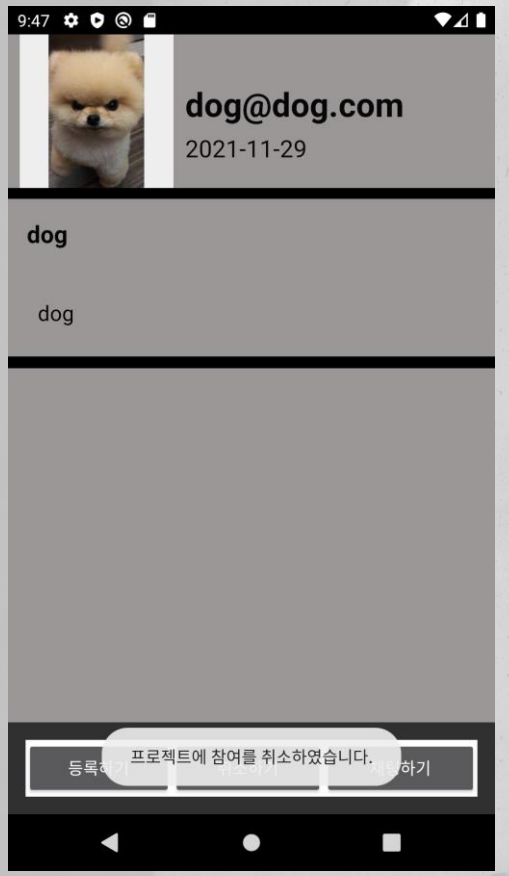
프로젝트 참가하기

```
reg_button.setOnClickListener(new View.OnClickListener() {
    @RequiresApi(api = Build.VERSION_CODES.N)
    @Override
    public void onClick(View view) {

        long now = System.currentTimeMillis();
        Date date = new Date(now);
        SimpleDateFormat sdf = new SimpleDateFormat( pattern: "yyyy-MM-dd");
        String getTime = sdf.format(date);
        String getId = firebaseUser.getEmail();
        Register userId = new Register(getId, getTime);
        FirebaseUser firebaseUser = mFirebaseAuth.getCurrentUser();
        idkey = mDatabase.child("register").child(key).push().getKey();
        mDatabase.child("register").child(key).child(idkey).setValue(userId);


        // 토스트메시지 출력
        Toast.makeText( context: ProjectActivity.this,  text: "프로젝트에 참여되었습니다.", Toast.LENGTH_SHORT).show();

        // 댓글 변환시 불러오는 함수
```

-해당 프로젝트에 참여하고 싶은 인원은 등록버튼을 누르면 됩니다.

프로젝트 취소하기

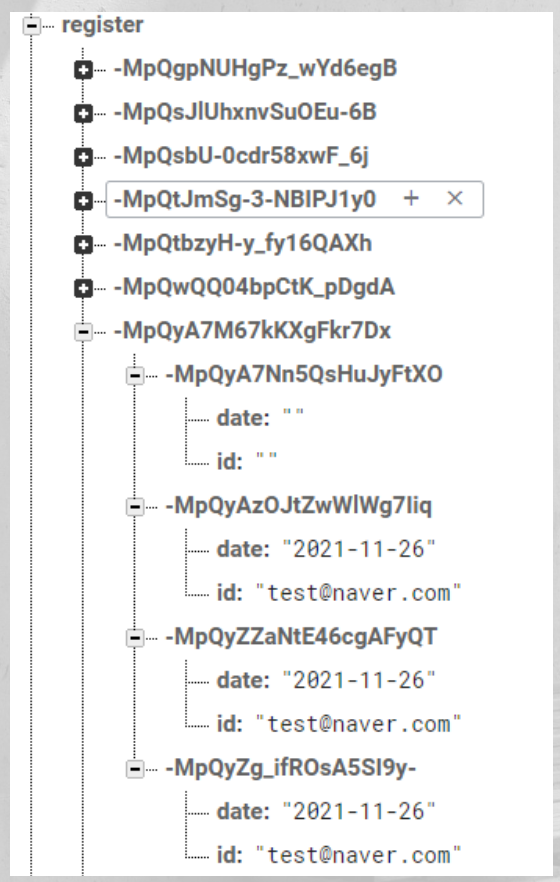




취소하기 버튼을 누르면 해당 프로젝트에서 글이 사라지게 됩니다.

프로젝트 취소하기

```
cancel_button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        DatabaseReference dataref = mDatabase.child("register").child(key).child(idkey);
        dataref.removeValue();

        Toast.makeText( context: ProjectActivity.this,  text: "프로젝트에 참여를 취소하였습니다.", Toast.LENGTH_SHORT).show();
    }
});
```



RemoveValue()함수를 호출하여 해당 정보를 firebase에서 제거합니다.

비대면 스터디 작성 게시판 WritingStudyActivity

Zoom 생성

등록하기를 누르면 게시물 작성자가 생성한 zoom에 참여할 수 있습니다.

Post

작성된 comment들을 화면에서 출력해줍니다. Firebase realtime data에 연결되어 있어 Array를 clear하고 onDataChange 함수를 통해 추가될 때마다 자동으로 출력합니다.

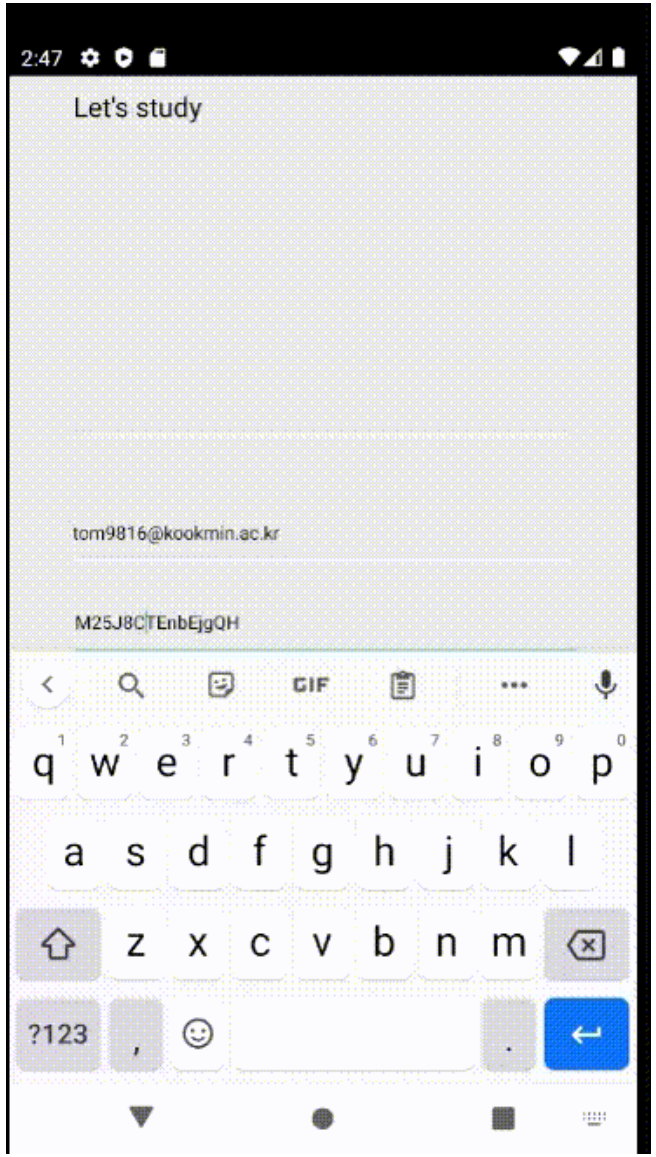
Zoom Id, Zoom Pwd

List를 클릭하면 해당하는 내용의 게시물의 내용이담긴 인덴스로 넘어가게 됩니다.

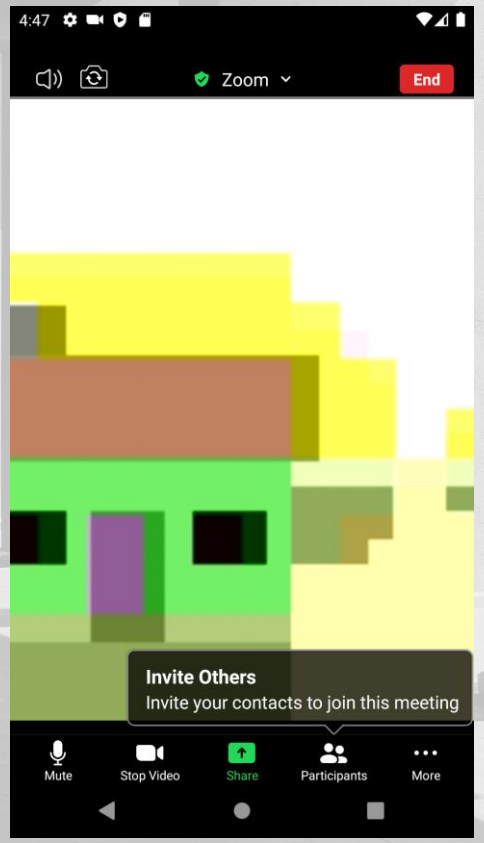
등록하기

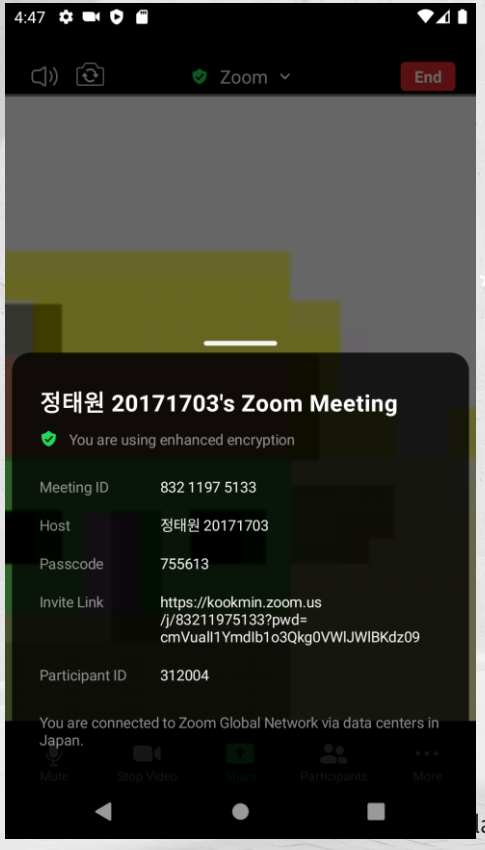
참여자로 리스트에 올라가게됩니다.

해당 영상은 줌 저작권 문제로
RECORD에 이상이 있습니다.

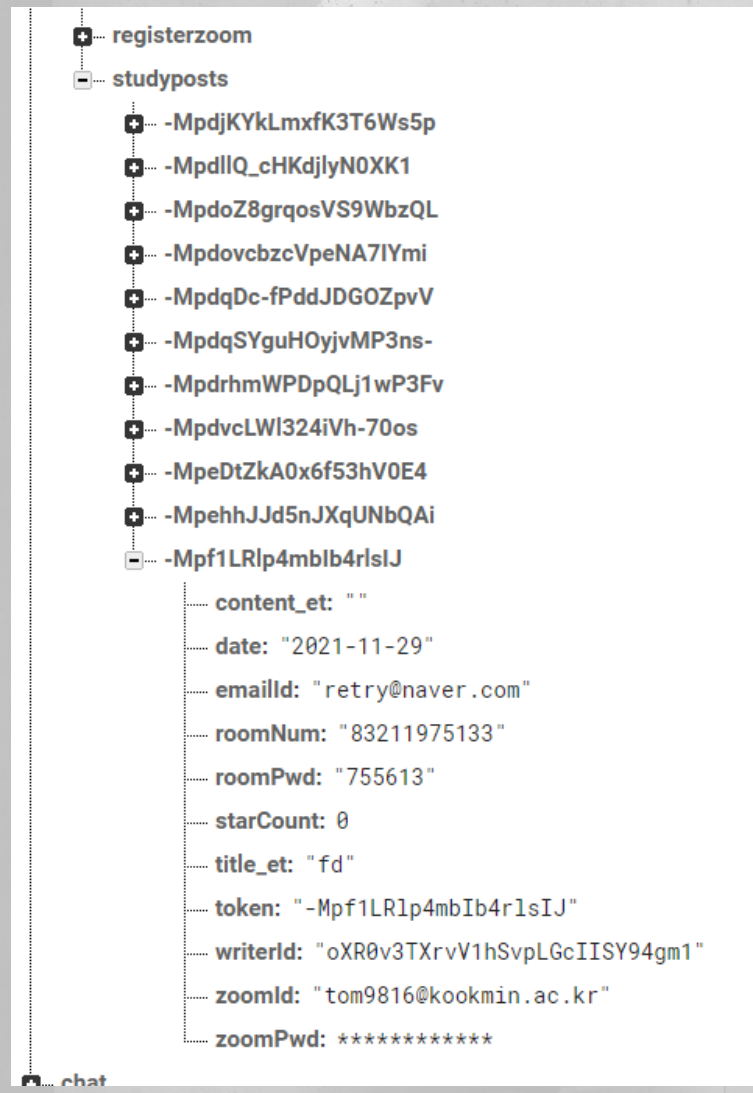


Zoom 생성


4:47
Zoom
End
Invite Others
Invite your contacts to join this meeting
Mute
Stop Video
Share
Participants
More


4:47
Zoom
End
정태원 20171703's Zoom Meeting
You are using enhanced encryption
Meeting ID 832 1197 5133
Host 정태원 20171703
Passcode 755613
Invite Link https://kookmin.zoom.us/j/83211975133?pwd=cmVuall1Ymdlb1o3Qkg0VWlJWlBKdz09
Participant ID 312004
You are connected to Zoom Global Network via data centers in Japan.

labs


registerzoom
studyposts
-MpdjKYkLmxfK3T6Ws5p
-MpdllQ_cHKdjlyN0XK1
-MpdoZ8grqosVS9WbzQL
-MpdovcbzcVpeNA7IYmi
-MpdqDc-fPddJDGOZpvV
-MpdqSYguHOyjvMP3ns-
-MpdrhmWPDpQLj1wP3Fv
-MpdvcLWl324iVh-70os
-MpeDtZkA0x6f53hV0E4
-MpehhJJd5nJXqUNbQAi
-Mpf1LRlp4mbIb4rlsIJ
content_et: ""
date: "2021-11-29"
emailId: "retry@naver.com"
roomNum: "83211975133"
roomPwd: "755613"
starCount: 0
title_et: "fd"
token: "-Mpf1LRlp4mbIb4rlsIJ"
writerId: "oXR0v3TXrvV1hSvpLGcIISY94gm1"
zoomId: "tom9816@kookmin.ac.kr"
zoomPwd: ***********
chat

```java
public void initializeSdk(Context context) {
    ZoomSDK sdk = ZoomSDK.getInstance();
    // TODO: Do not use hard-coded values for your key/secret in your app in production!
    ZoomSDKInitParams params = new ZoomSDKInitParams();
    params.appKey = "8ldKjhRAXw7nkMcOW6UIJtU7UbIFBfBPOKfb"; // TODO: Retrieve your SDK key and enter it here
    params.appSecret = "nLu6cbw9EeJWNtH2g2RU2ftziQXVQ4W4jOet"; // TODO: Retrieve your SDK secret and enter it here
    params.domain = "zoom.us";
    params.enableLog = true;
    // TODO: Add functionality to this listener (e.g. logs for debugging)
    ZoomSDKInitializeListener listener = new ZoomSDKInitializeListener() {
        /**
         * @param errorCode {@link us.zoom.sdk.ZoomError#ZOOM_ERROR_SUCCESS} if the SDK has been initialized successfully.
         */
        @Override
        public void onZoomSDKInitializeResult(int errorCode, int internalErrorCode) { }

        @Override
        public void onZoomAuthIdentityExpired() { }
    };
    sdk.initialize(context, listener, params);
```

Zoom Sdk에 appkey와 secret을 입력하여 사용할 준비를 합니다. Firebase에서 게시물의 내용을 가져와 회의실 ID와 Password를 모듈에 입력으로 주고 Zoom sdk 모듈에서 이를 실행하게 합니다.

-

```java
public void startMeeting(Context context) {

    ZoomSDK sdk = ZoomSDK.getInstance();
    MeetingServiceListener meetingServiceListener = new MeetingServiceListener(){

        @Override
        public void onMeetingStatusChanged(MeetingStatus meetingStatus, int i, int i1) {
            if(meetingStatus == meetingStatus.MEETING_STATUS_INMEETING) {
                meetingID = String.valueOf(sdk.getInMeetingService().getCurrentMeetingNumber());
                meetingPwd = sdk.getInMeetingService().getMeetingPassword();
                FirebaseUser firebaseUser = mFirebaseAuth.getCurrentUser();
                PostZoom post = new PostZoom();

                post.setEmailId(firebaseUser.getEmail());
                post.setTitle_et(strTitle);
                post.setContent_et(strContent);
                post.setZoomId(strZoomId);
                post.setZoomPwd(strZoomPwd);
                post.setDate(getTime);
                post.setWriterId(firebaseUser.getUid());
                post.setRoomNum(meetingID);
                post.setRoomPwd(meetingPwd);
                String key = mDatabaseRef.child("studyposts").push().getKey();
                post.setToken(key);
                Register regi = new Register();
                mDatabaseRef.child("studyposts").child(key).setValue(post);
                mDatabaseRef.child("registerzoom").child(key).push().setValue(regi);

                mDatabaseRef.child("UserAccount").child(firebaseUser.getUid()).child("postzoom").push().setValue(key);
            }
        }
```

Zoom 모듈인 startMeeting을 커스텀하여 회의가 열리는 동시에 firebase에 해당 글과 회의실 아이디 및 비밀번호가 저장되게 됩니다. 이글은 추후에 참여자가 입장할 때 사용합니다.

-

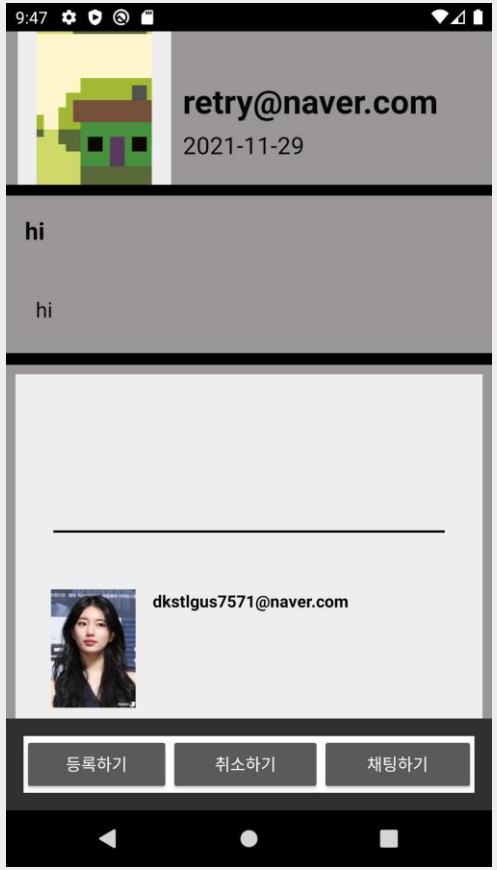
비대면 스터디 게시판 StudyActivity

Zoom 참여기능

등록하기를 누르면 게시물 작성자가 생성한 zoom에 참여할 수 있습니다.

참여자 리스트 확인

작성된 comment들을 화면에서 출력해줍니다. Firebase realtime data에 연결되어 있어 Array를 clear하고 onDataChange 함수를 통해 추가될 때 마다 자동으로 출력합니다.



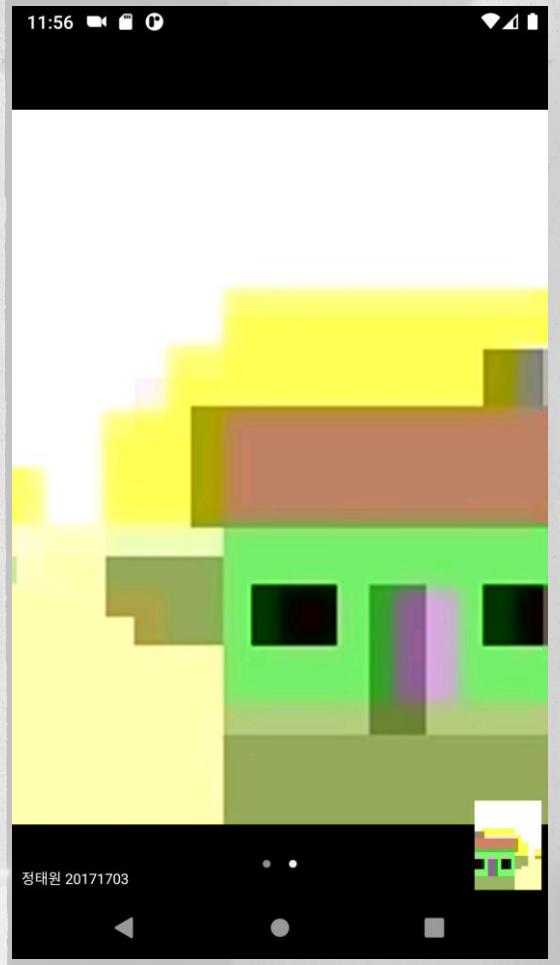
채팅방 기능

List를 클릭하면 해당하는 내용의 게시물의 내용이담긴 인덴스로 넘어가게 됩니다.

등록하기

참여자로 리스트에 올라가게됩니다.

Zoom 참여기능



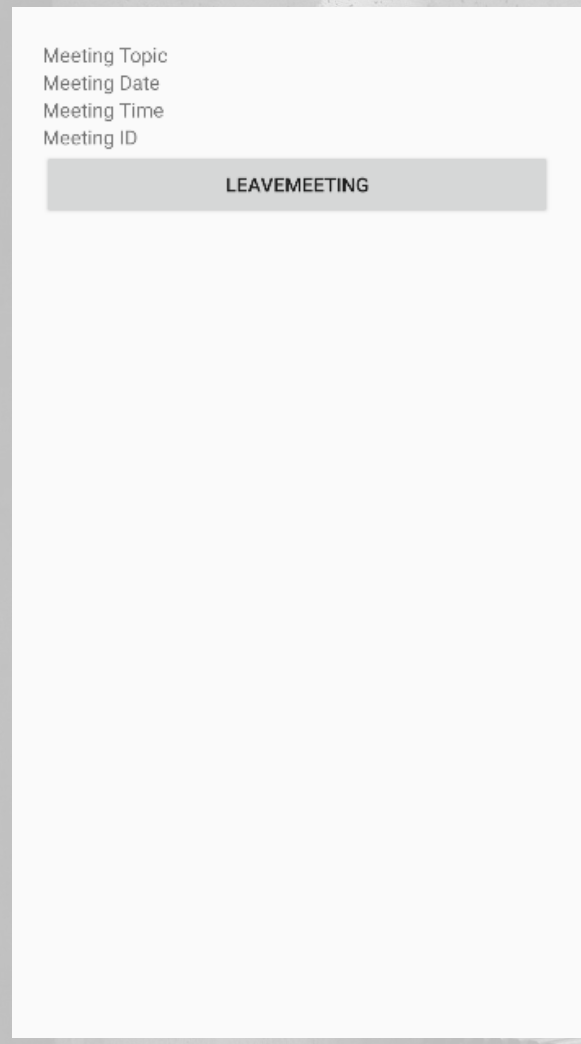
```
private void createJoinMeetingDialog() {
    if (this.userid != null && this.meetingno != null &&  this.meetingpwd != null ) {
        String Name =userid;
        String meetingNumber = meetingno;
        String password = meetingpwd;
        if (meetingNumber.trim().length() > 0 && password.trim().length() > 0) {
            joinMeeting( context: ZoomActivity.this, Name, meetingNumber, password);
        }
    }
}
```

```
public void joinMeeting(Context context, String Name, String meetingNumber, String password) {
    MeetingService meetingService = ZoomSDK.getInstance().getMeetingService();
    JoinMeetingOptions options = new JoinMeetingOptions();
    JoinMeetingParams params = new JoinMeetingParams();
    params.displayName = Name;
    params.meetingNo = meetingNumber;
    params.password = password;
    meetingService.joinMeetingWithParams(context, params, options);
}
```

Zoom에 참여하는 모듈입니다. Firebase에서 게시물의 내용을 가져와 회의실 ID와 Password를 모듈에 입력으로 주고 Zoom sdk 모듈에서 이를 실행하게 합니다.

-

Zoom 회의실 기다리기



```xml
<activity
    android:name=".WaitActivity"
    android:exported="true">
    <intent-filter>
        <action android:name="com.terminaldevelopers.myzoom.intent.action.JoinBeforeHost"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
</activity>
```

```java
LeaveMeeting.setOnClickListener(this);
ZoomSDK zoomSDk = ZoomSDK.getInstance();
MeetingService meetingService = zoomSDk.getMeetingService();
if(meetingService !=null){
    meetingService.addListener( meetingServiceListener: this);
}
```

Zoom에 참여하는 모듈입니다. Firebase에서 게시물의 내용을 가져와 회의실 ID와 Password를 모듈에 입력으로 주고 Zoom sdk 모듈에서 이를 실행하게 합니다. AddListener 는 meetingService의 상태를 파악하여 zoom을 불러옵니다.

-

E N D O F D O C U M E N T

THANK YOU

PAPATA LABS