

# ComPlant 최종 발표

반려 식물 일지 애플리케이션  
컴퓨터공학과 20174706 김혜민  
컴퓨터공학과 20176189 최정원

# 목차

01

중간 발표 내용  
요약

02

기능 설명

03

구현 과정

04

주요 소스코드

05

향후 계획 및  
느낀 점

06

실행 화면  
동영상

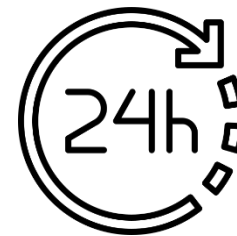
## 01. 중간 발표 내용 요약

## ComPlant=>companion plant



### 사용자

- 취미생활로 반려 식물을 기르고자 하는 사람
- 식물을 기르는데 초보자인 사람
- 반려 식물의 성장을 기록하고자 하는 사람
- 취미생활을 더욱 즐겁게 하고자 하는 사람



### 사용 시간

- 일상생활 속 힐링이 필요한 시간



### 장소

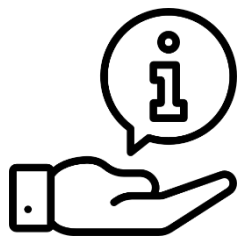
- 집, 사무실, 책상 위, 화단 등  
개인공간이라면 어디든 가능



### 목적

- 최근 식물 기르는 취미 인기
- 식물 정보 공유 플랫폼
- 반려 식물과의 교감

## 기대효과



### 정보제공

- 식물 사전 기능을 통한 정보제공



### 힐링 공간

- 메인페이지에 설정 메시지를 주기적으로 보여 줌으로써 '힐링 공간'이라는 특성 강조



### 커뮤니티

- 게시판을 활용한 커뮤니티 활성화

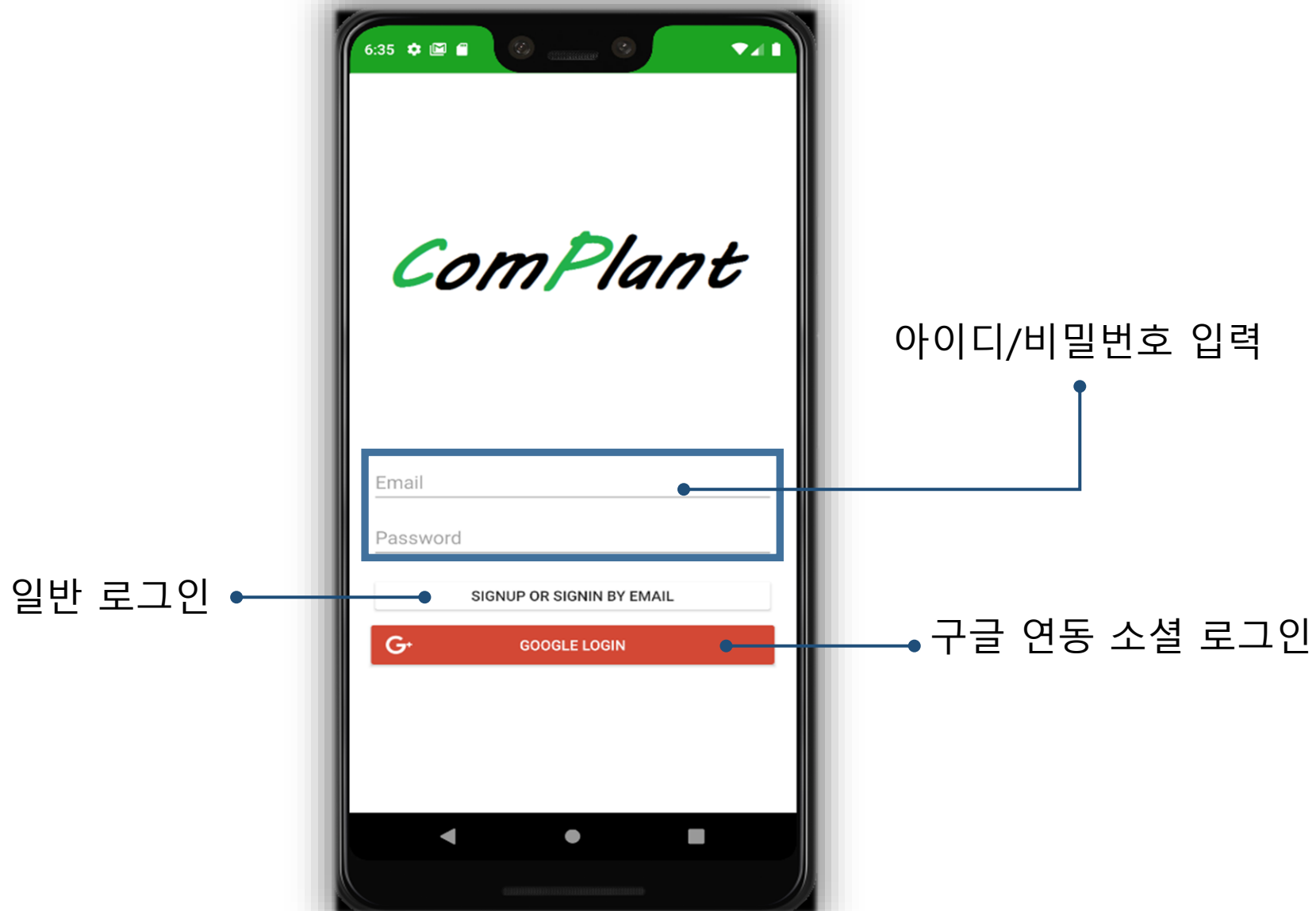


### 직관성

- 기존 서비스의 단점은 직관성이 부족하다는 점
- 적은 기능으로 구성되어있음에도 불편함
- ComPlant는 사용자 편의성을 위해 직관적인 레이아웃 구성

## 02. 기능 설명

## 일반 로그인, 소셜 로그인



## (좌) 홈, (우) 마이페이지

마이페이지로 이동

설정  
한  
프로필  
사진

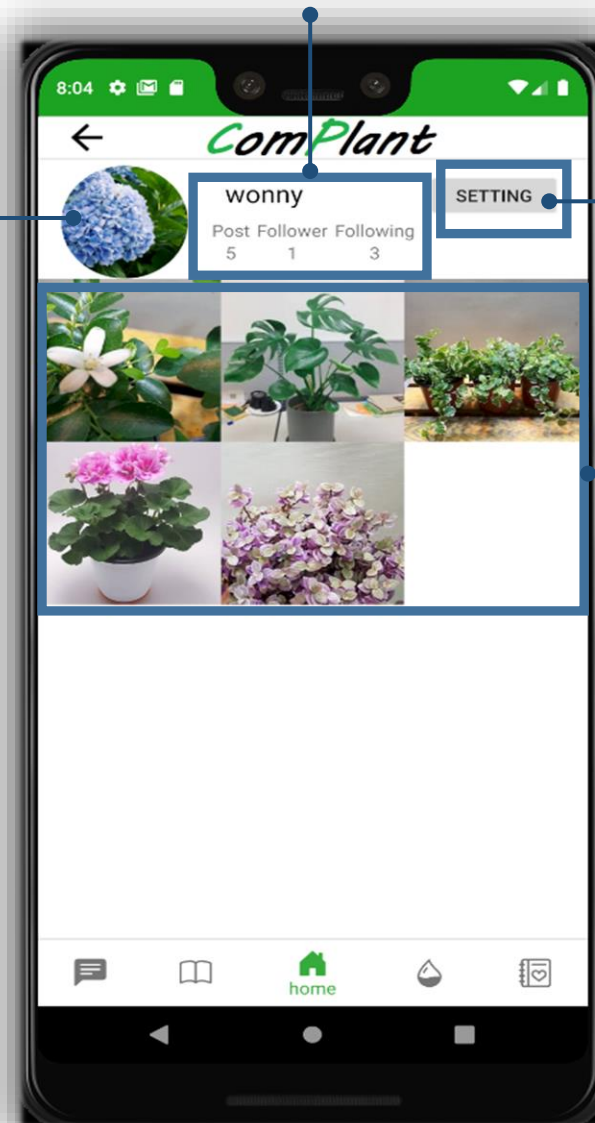


클릭 시 프로필 변경  
설정으로 이동

설정  
한  
메시지  
내용

사용자 이름과  
식물 별칭

- 사용자 이름
- 게시글 수
- Follower/Following 수

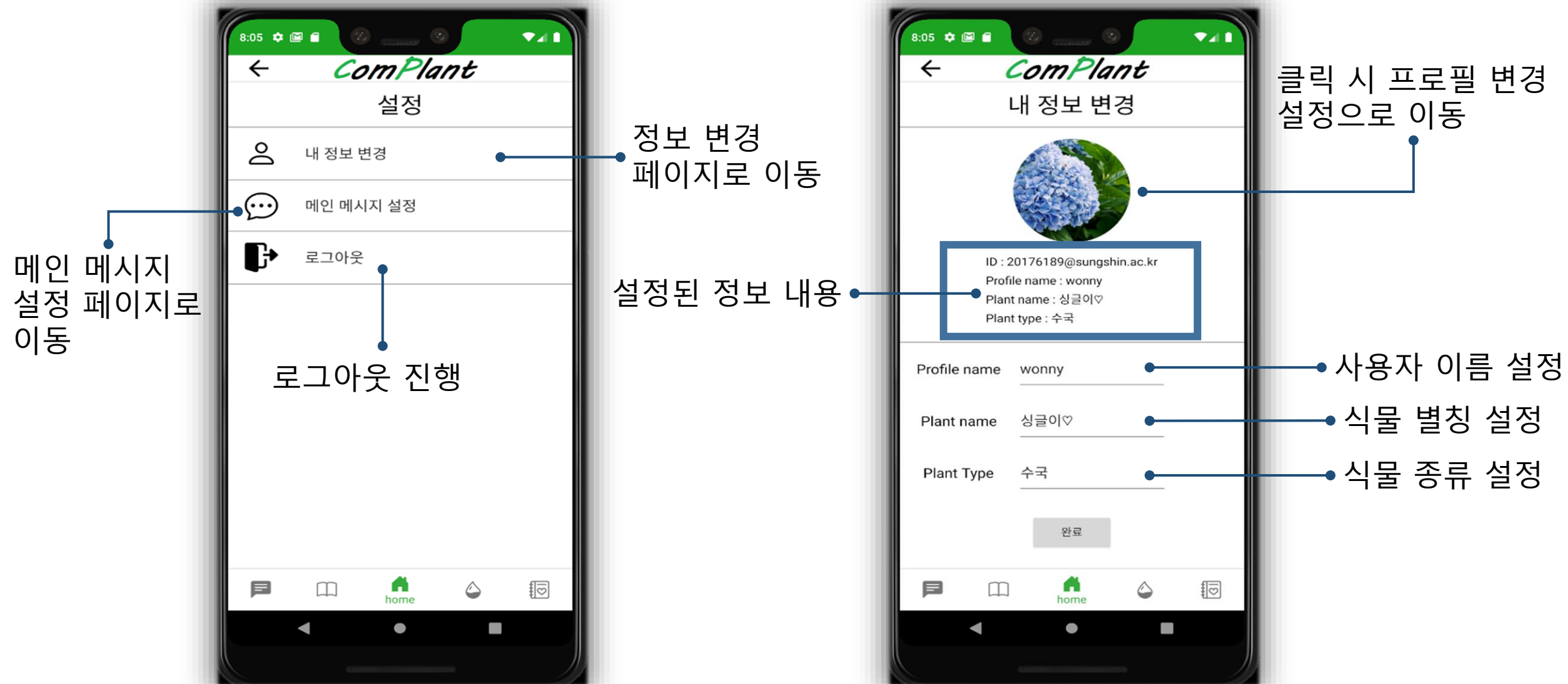


설정  
페이지로  
이동

해당 식물 일지  
게시글로 이동

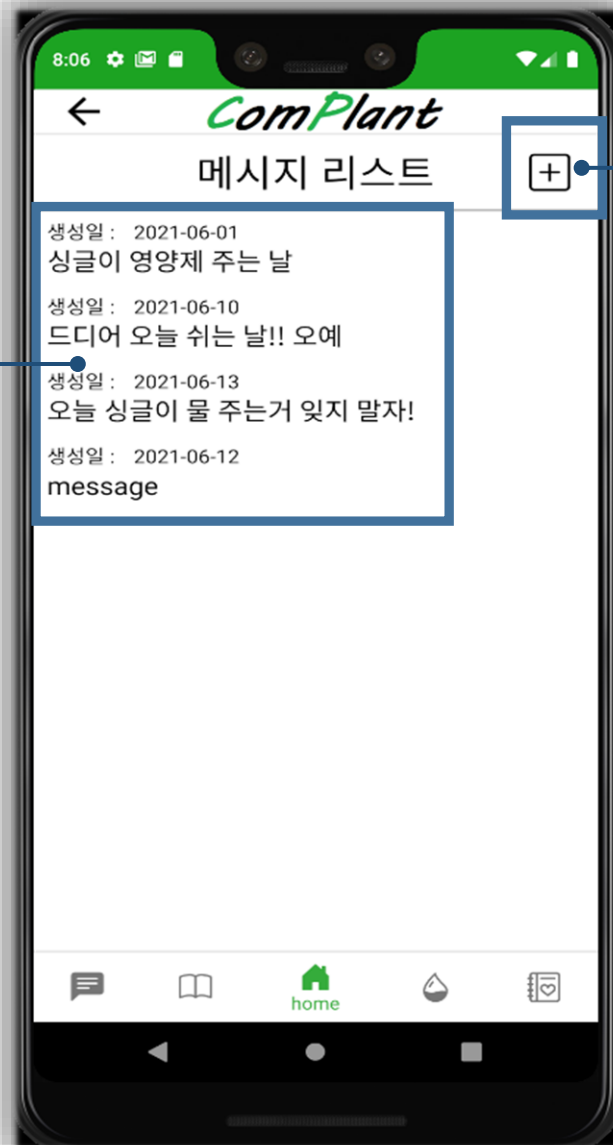


## (좌) 설정 화면, (우) 내 정보 변경



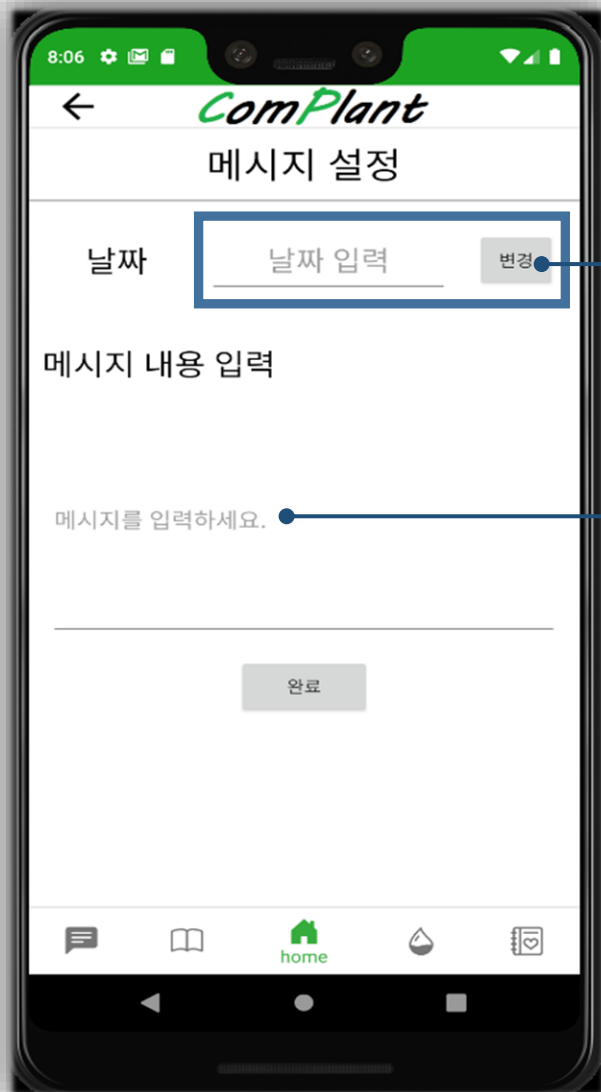
## 메시지 리스트

설정 완료된 메시지 리스트



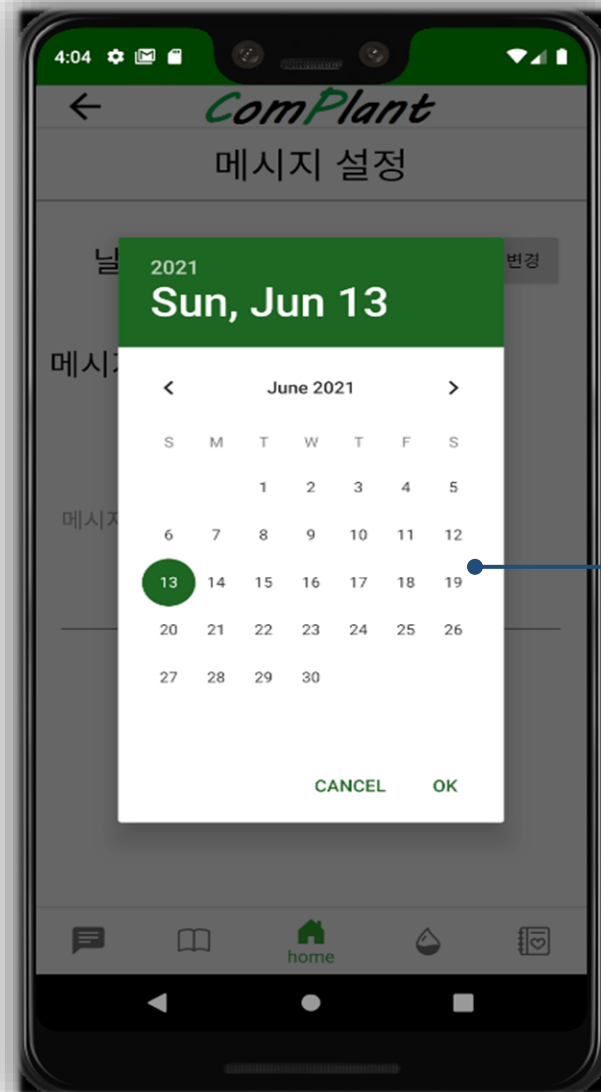
메시지 리스트 설정 페이지로 이동

## 메시지 설정



변경 버튼 클릭  
후 생성일 입력

홈에 띄울  
메시지 입력



변경 버튼 클릭 시  
달력에서 생성일  
선택 가능

## 게시판 메인



게시글 작성 페이지로 이동

해당 계정의 페이지로 이동

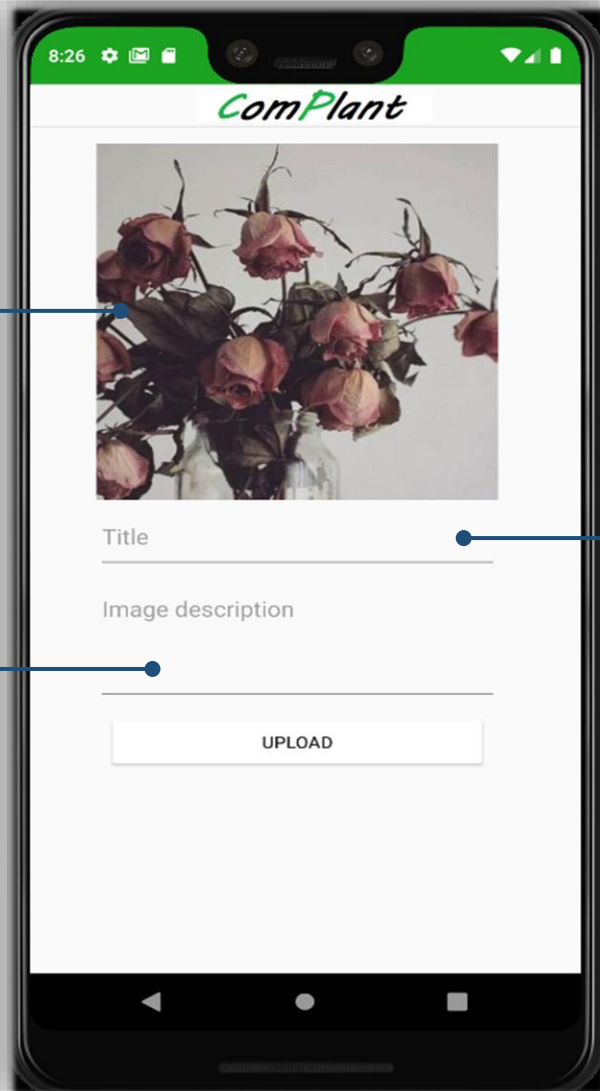
댓글 작성 페이지로 이동

공감 수 표시

게시글 작성자 표시

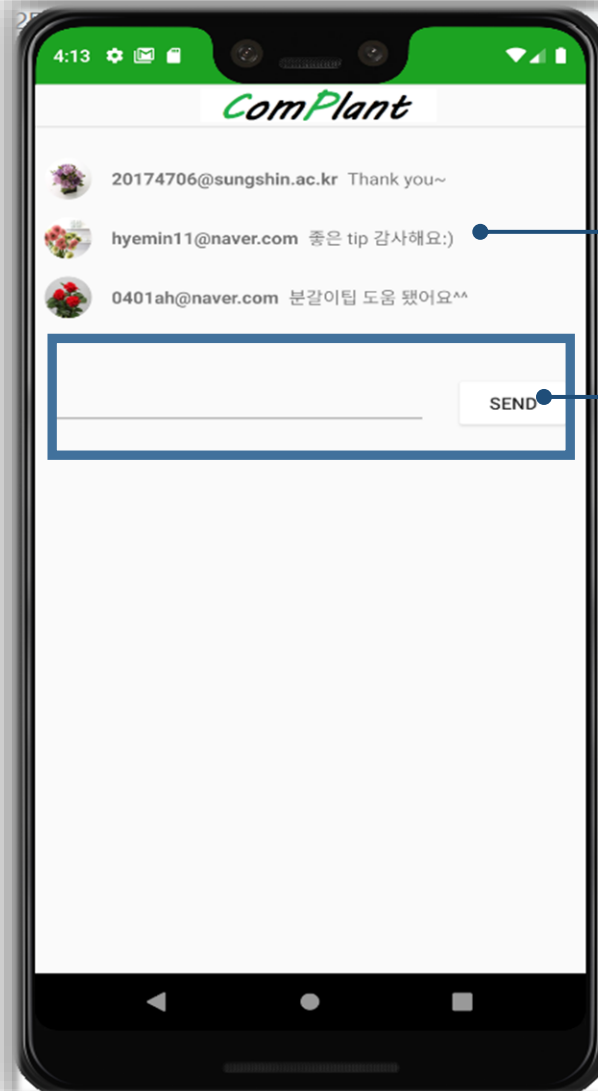
## (좌) 게시물 작성, (우) 게시판 댓글

게시 할 사진  
표시



게시글 내용

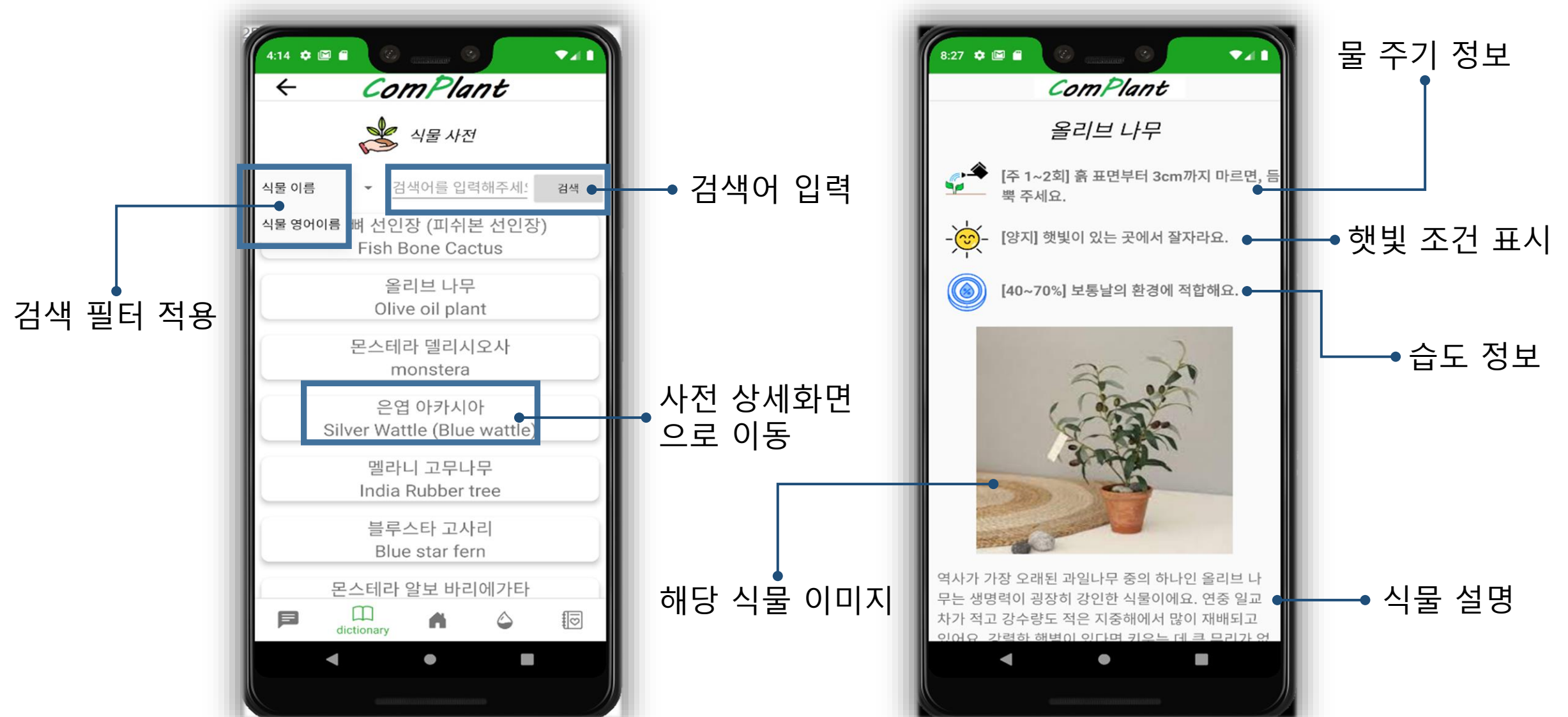
게시글 제목



작성된 댓글 표시

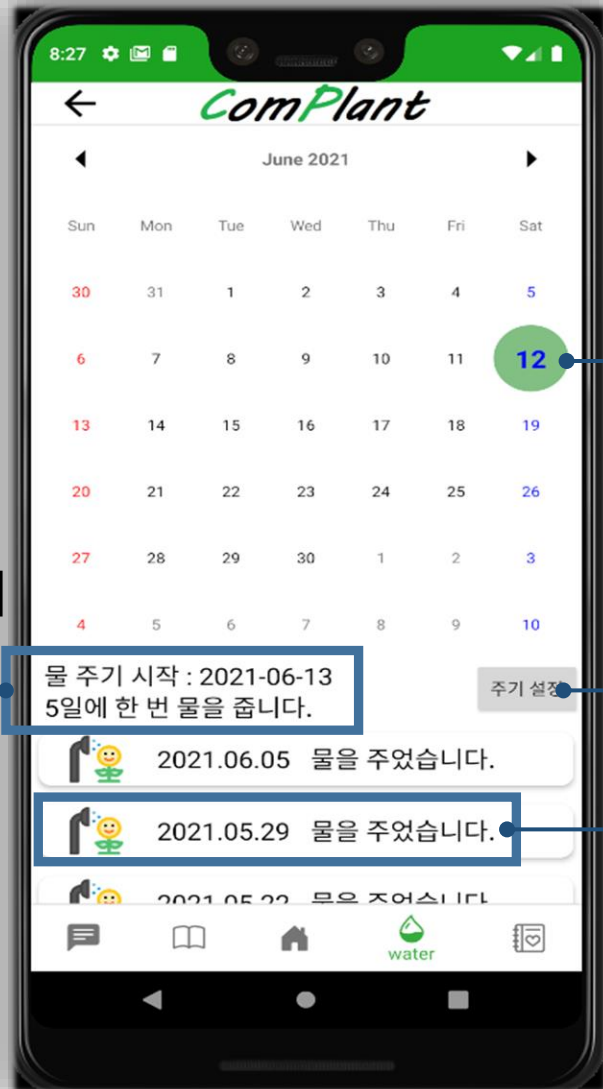
댓글 작성

## (좌) 식물 사전 메인, (우) 식물 사전 상세



## (좌) 물 주기 메인, (우) 물 주기 체크

물 주기 시작 날짜, 물 주기 표시

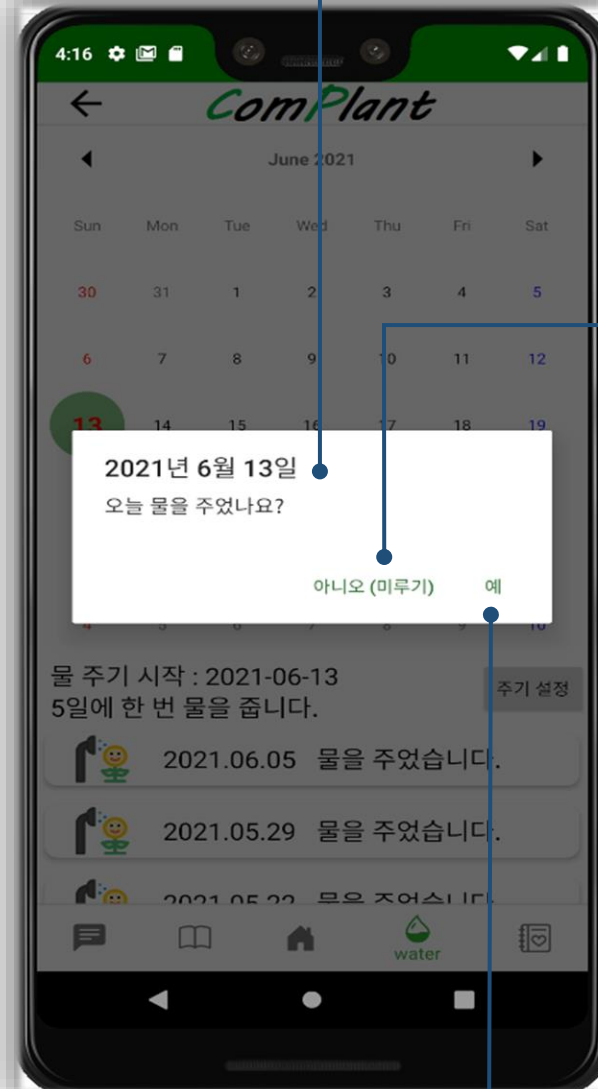


오늘 날짜 클릭 시 물을 주었는지 체크

물 주기 변경 페이지로 이동

'예' 클릭 시 오늘 날짜로 리사이클러뷰 출력

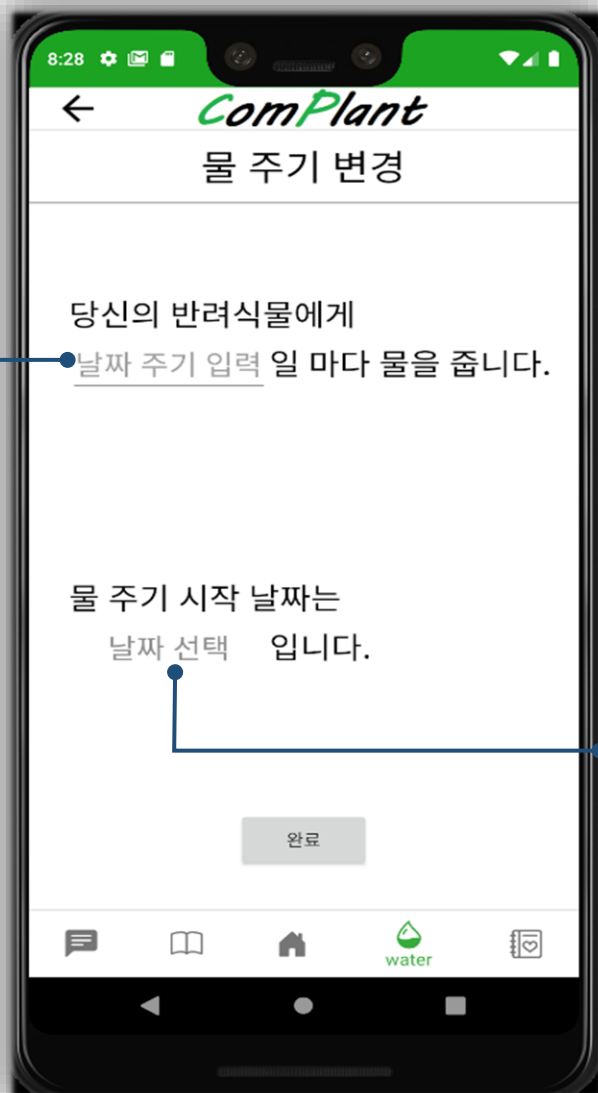
오늘 날짜 클릭 시 나타나는 대화상자



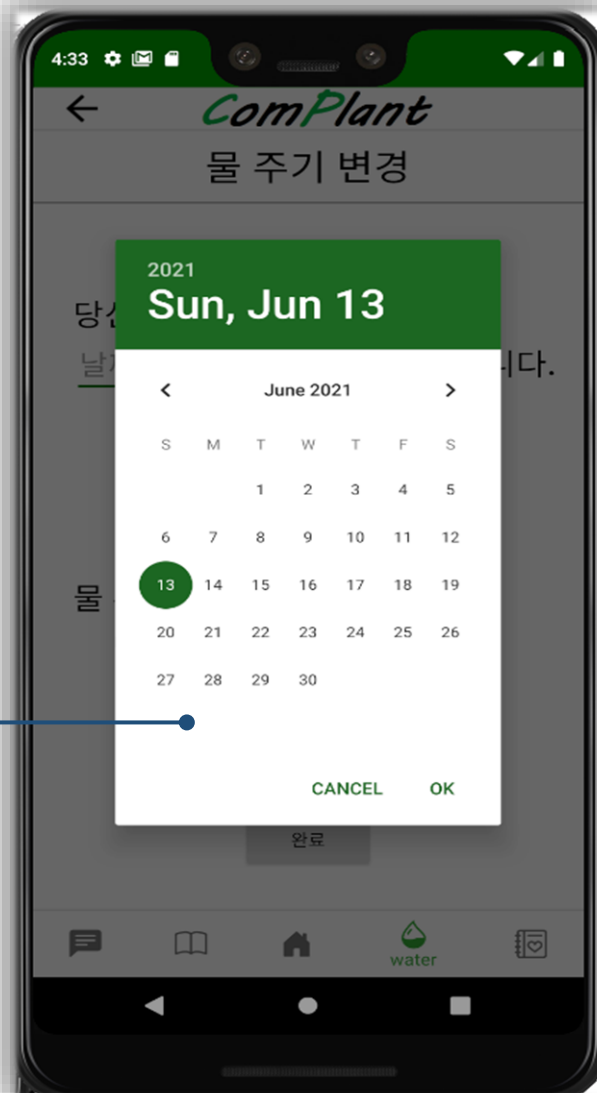
'아니오 (미루기)' 클릭 시 물 주기 변경 페이지로 이동

## 물 주기 변경

날짜 주기 입력

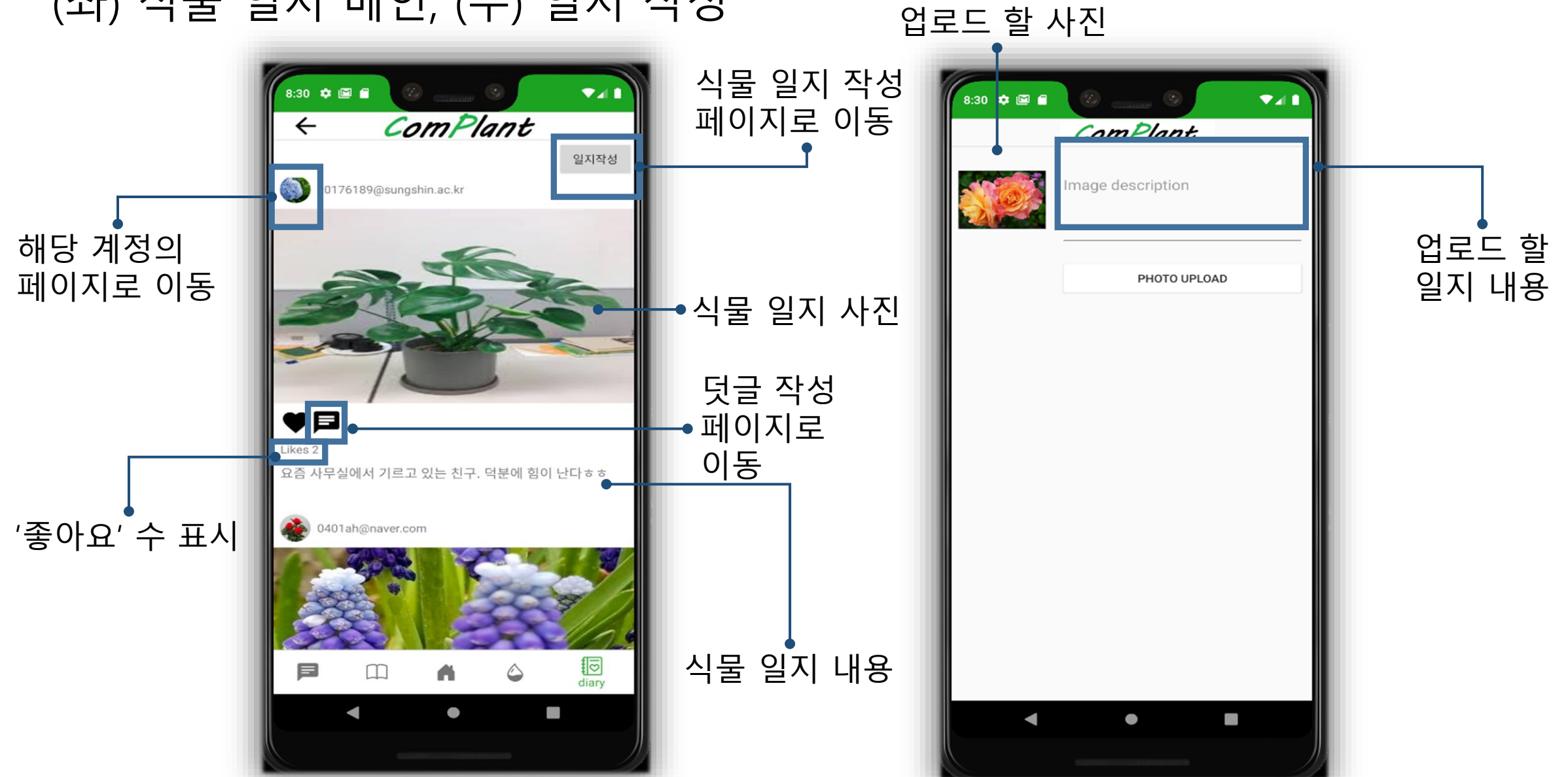


날짜 선택 클릭  
시 달력으로 날짜  
선택





## (좌) 식물 일지 메인, (우) 일지 작성

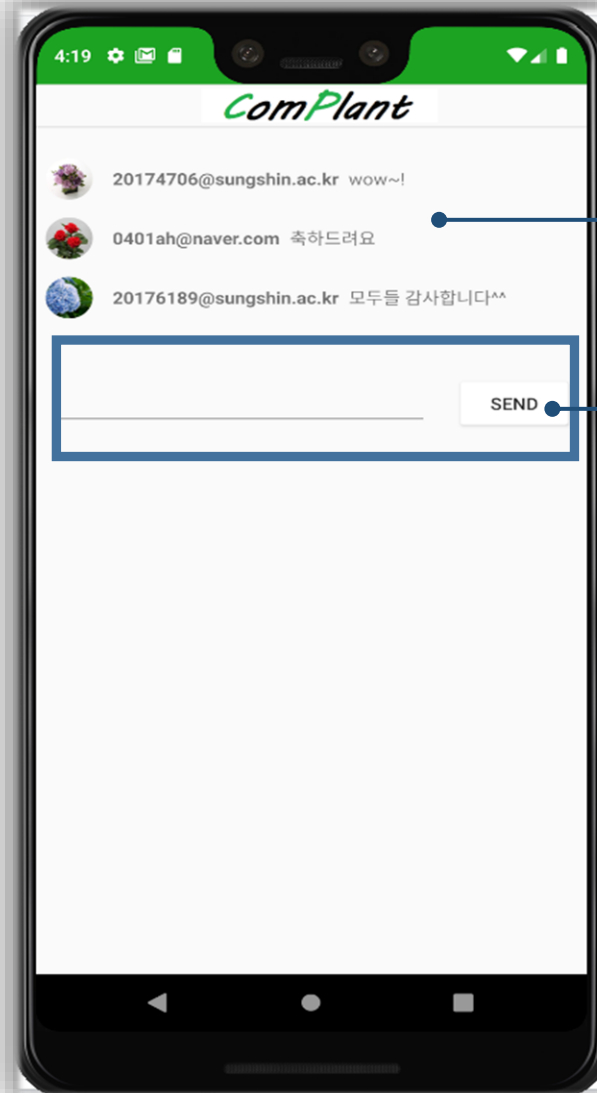


## (좌) 다른 계정의 마이페이지, (우) 일지 댓글

해당 식물 일지  
게시글로 이동



이웃 신청



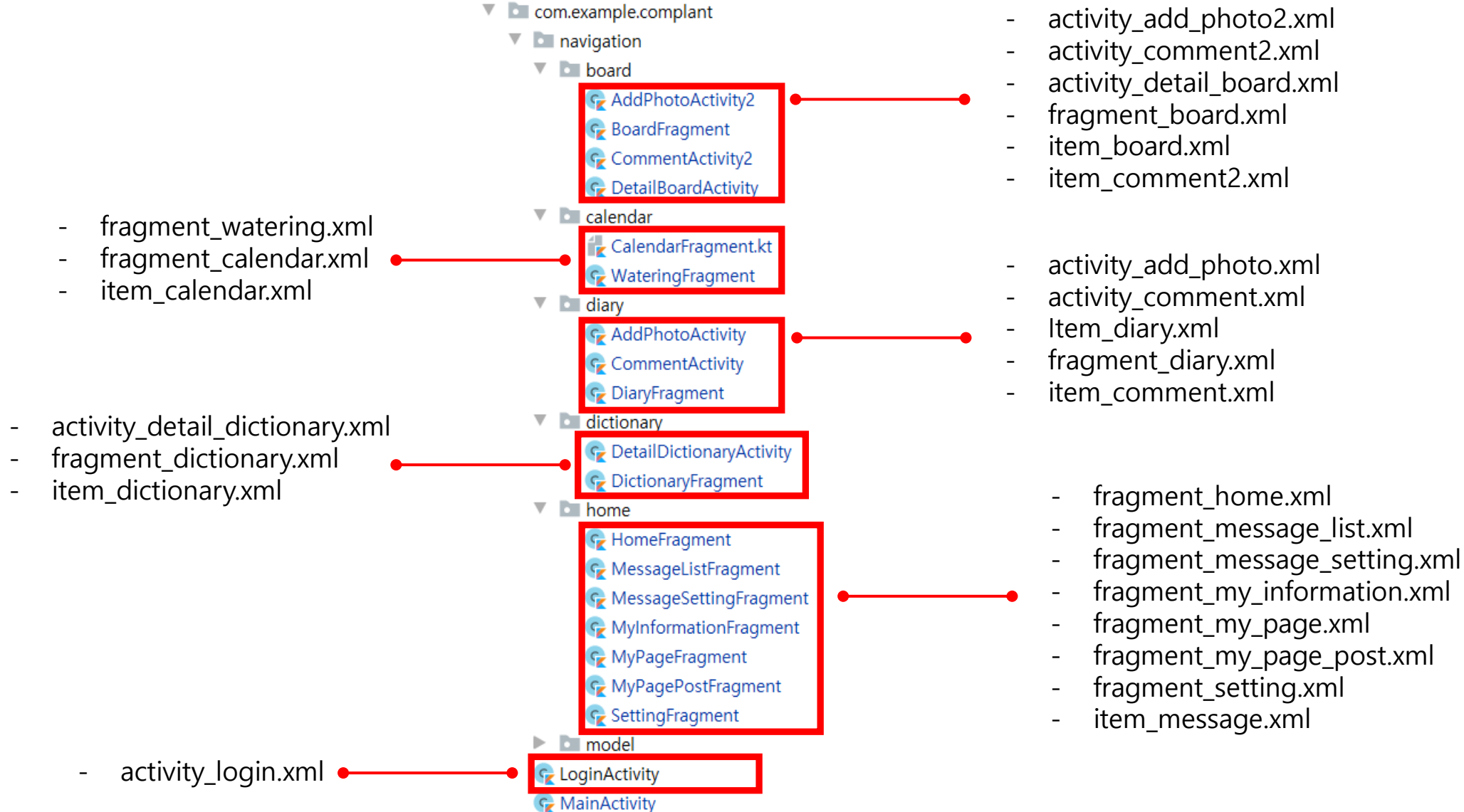
작성된 댓글 표시

댓글 작성

## 03. 구현 과정

- 모듈 구성도
- DB 설계
- 일정표

# 모듈 구성도



## 데이터 모델링(식물 일지 예시)

```
// explain -> content 의 설명을 관리
// imageUrl -> 이미지 주소 관리
// uid -> (어느 유저가 올렸는지) 유저 아이디를 관리
// userId -> 올린 유저의 이메일을 관리
// timestamp -> 언제 content 를 올렸는지 관리
// favoriteCount -> '좋아요'를 몇개 눌렀는지 관리
// favorites -> '중복 좋아요'를 방지할 수 있도록 '좋아요'를 누른 유저를 관리할 수 있는 Map

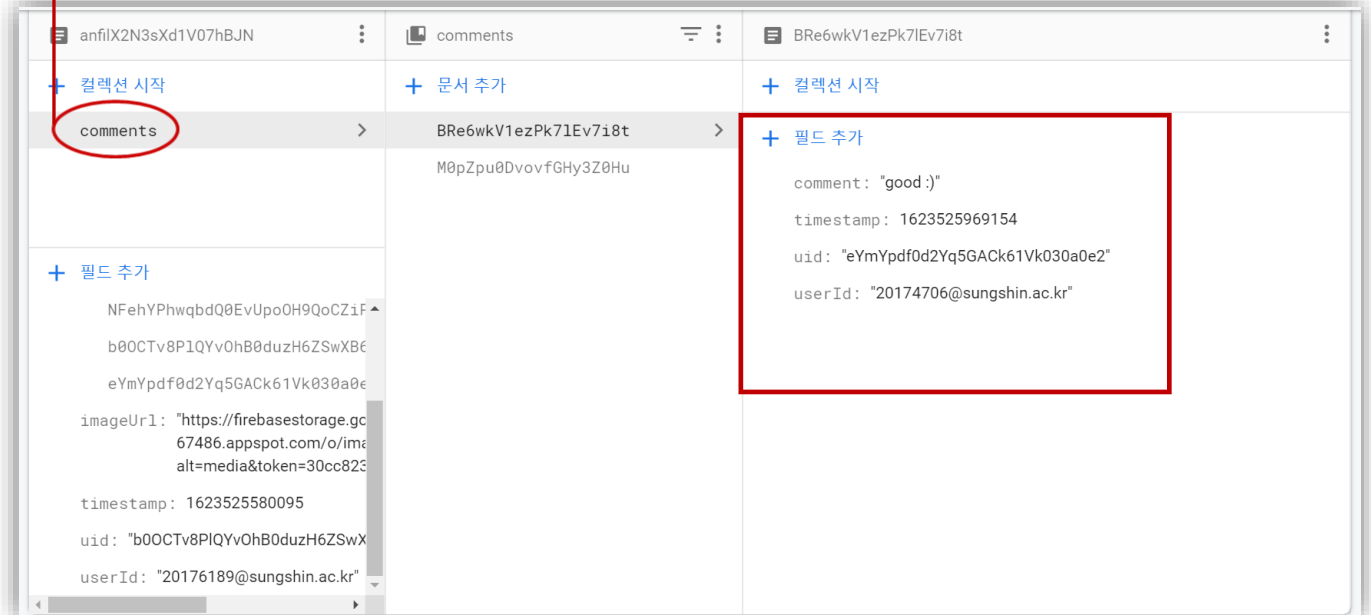
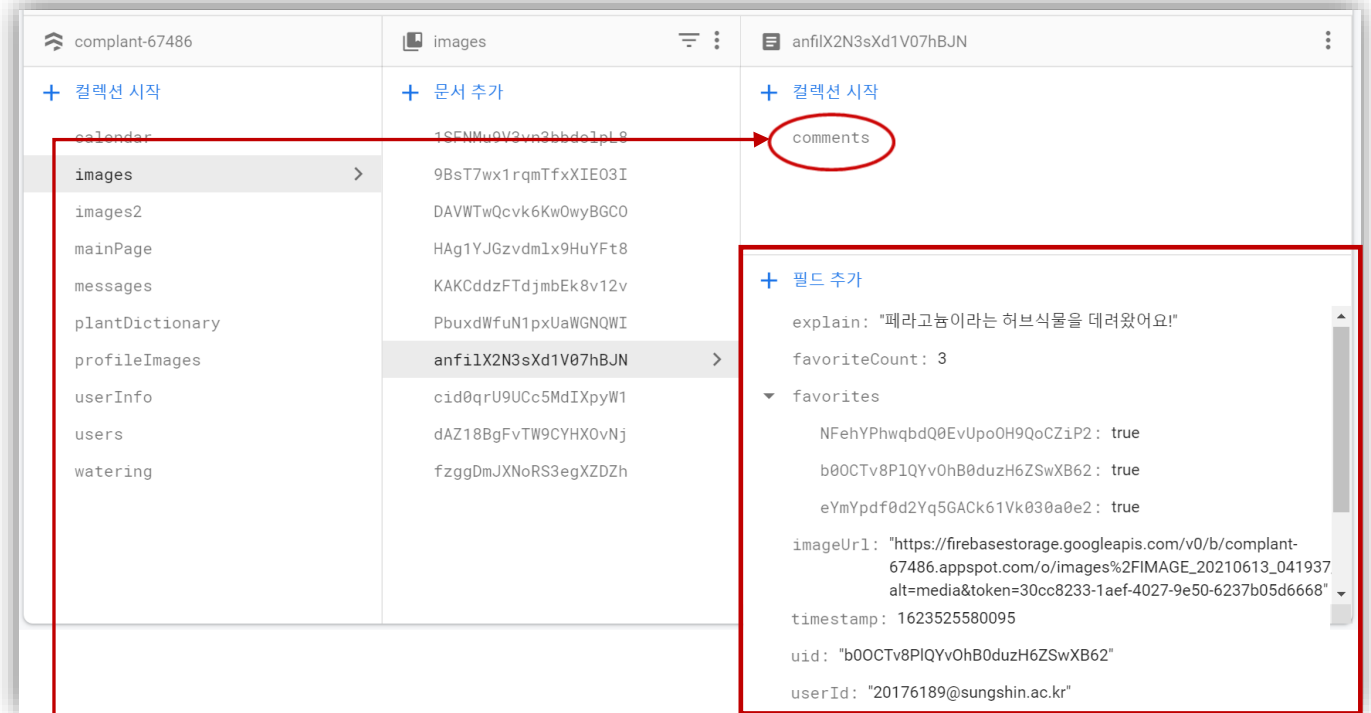
// Comment -> 댓글을 관리하는 data class

data class ContentDTO(
    var explain : String? = null,
    var imageUrl : String? = null,
    var uid : String? = null,
    var userId : String? = null,
    var timestamp : Long? = null,
    var favoriteCount : Int = 0,
    var favorites : MutableMap<String, Boolean> = HashMap()
){
    data class Comment(
        var uid : String? = null,
        var userId : String? = null,
        var comment : String? = null,
        var timestamp : Long? = null
    )
}
```

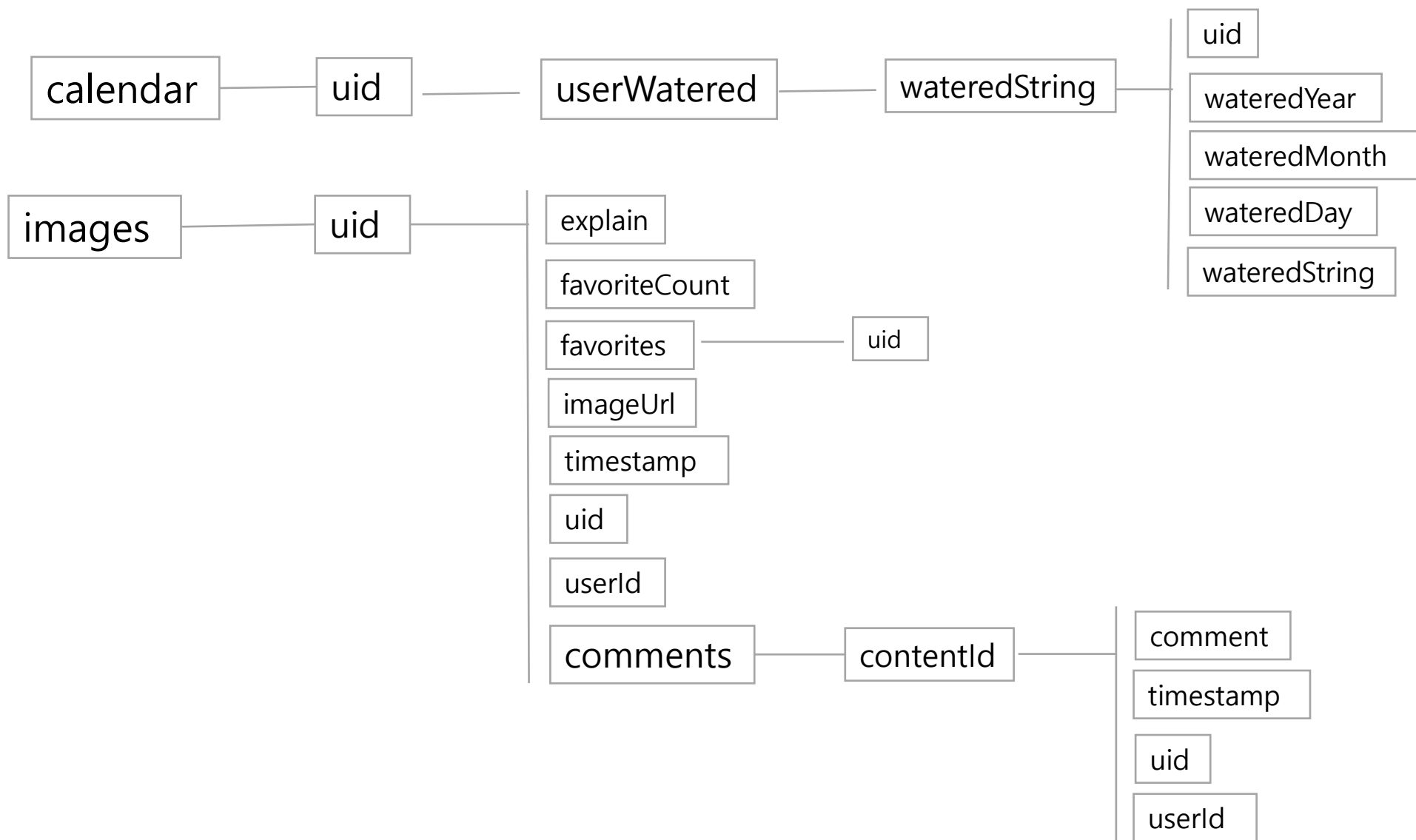
ContentDTO.kt

# 데이터 모델링(식물 일지 예시)

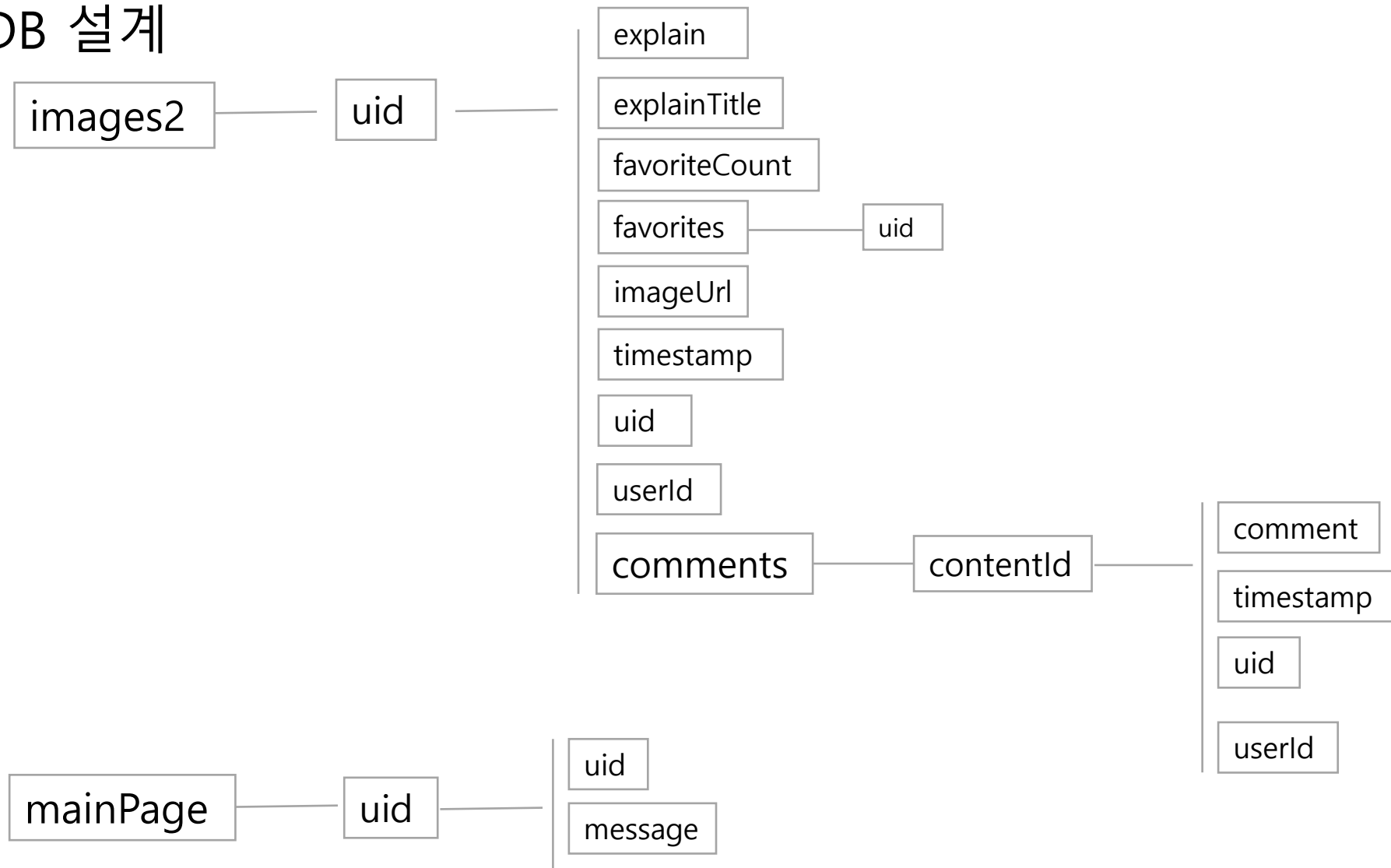
ContentDTO를 뼈대 삼아 만든 컬렉션 'image'



## DB 설계

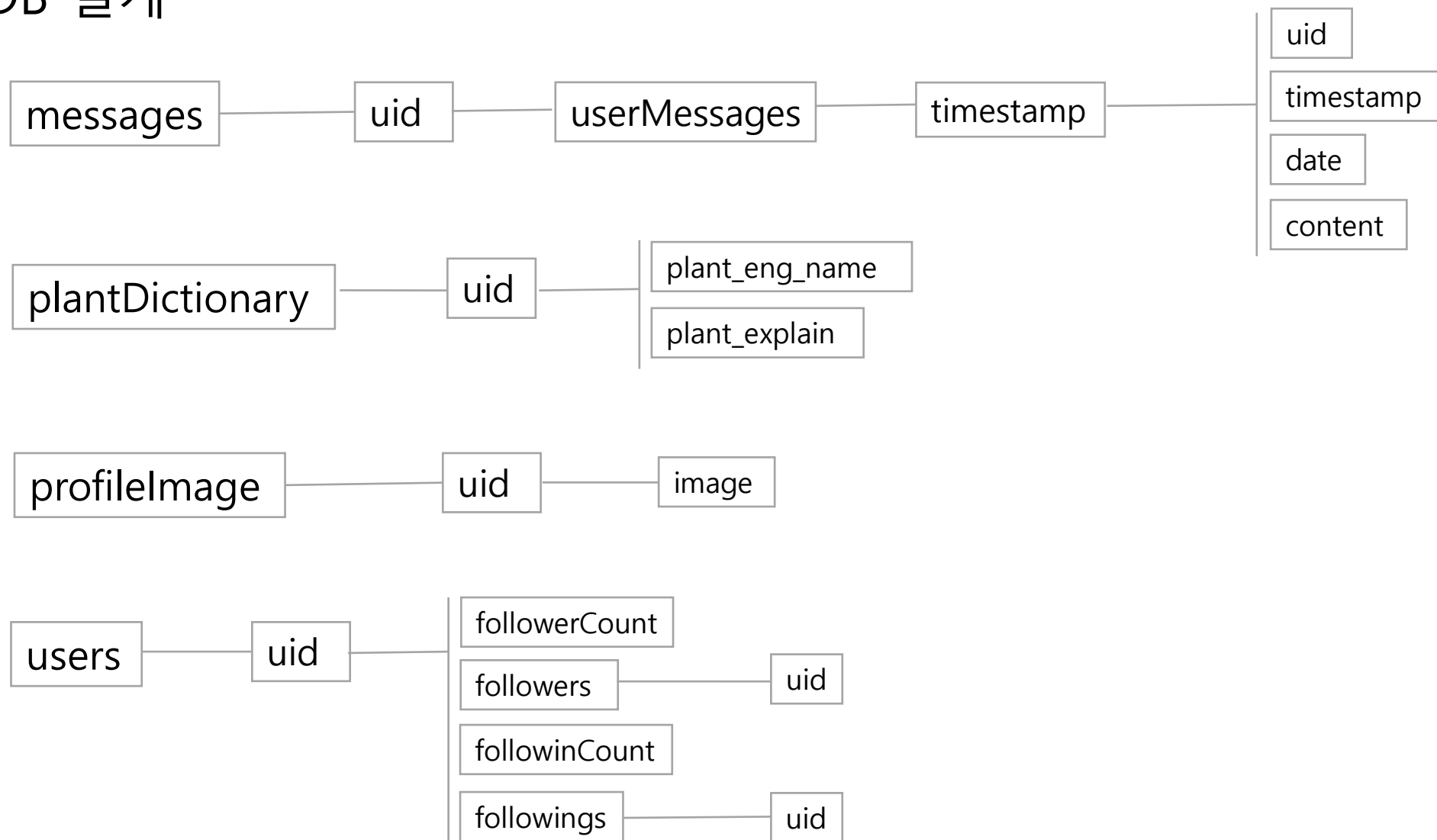


## DB 설계

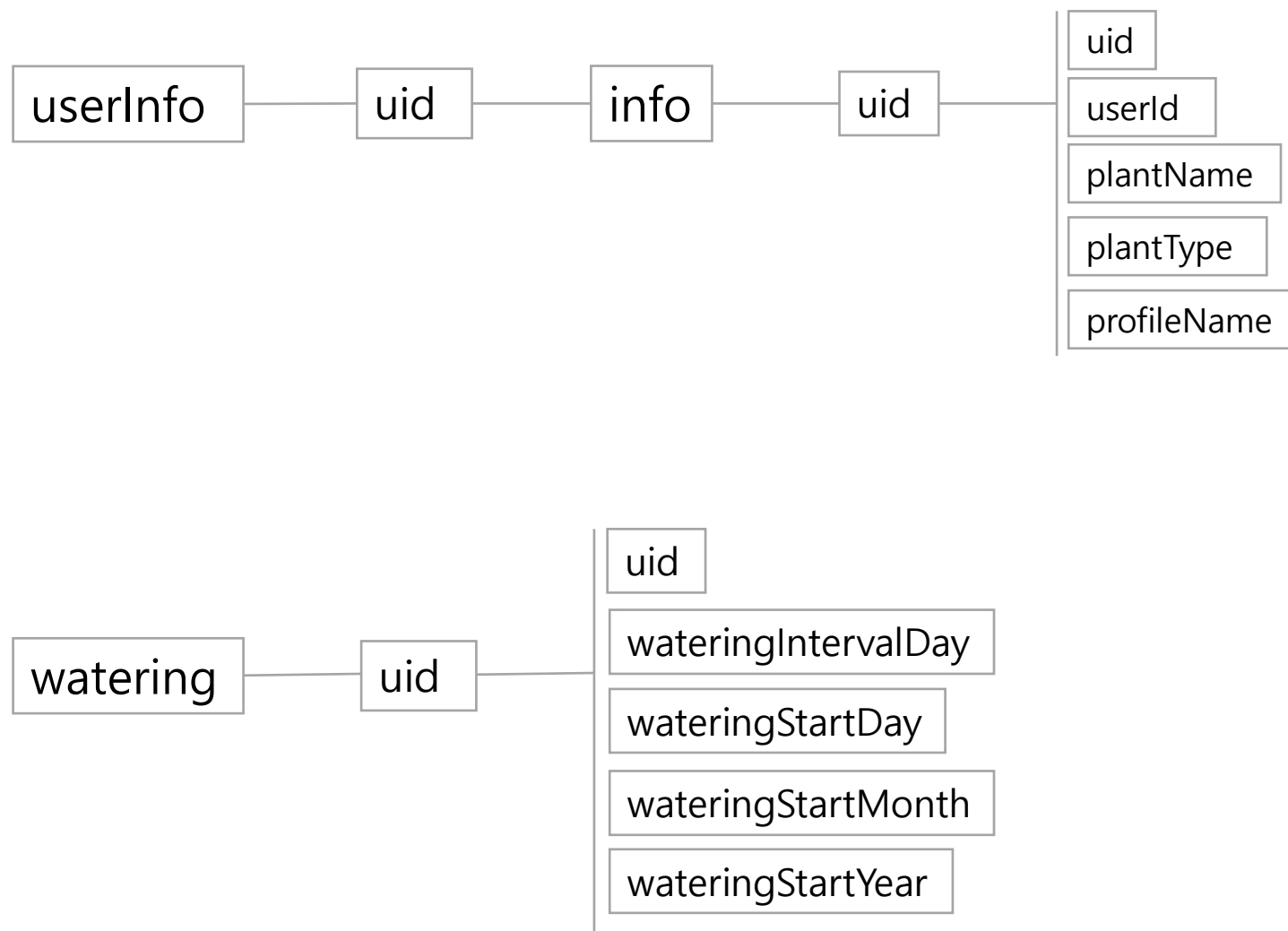




## DB 설계



## DB 설계



# 중간 발표 이후 일정

구분	8주차 (4/26~5/2)	9주차 (5/3~5/9)	10주차 (5/10~5/16)	11주차 (5/17~5/23)	12주차 (5/24~5/30)	13주차 (5/31~6/6)	14주차 (6/7~6/13)
김혜민	전체 내비게이션 바 레이아웃	홈 프래그먼트 레이아웃	마이페이지 레이아웃	setting 레이아웃	메인 메시지 레이아웃	물 주기 설정 레이아웃	레이아웃, 기능 수정 및 발표준비
	내비게이션 작동 개발	홈 프래그먼트 코드 개발	마이페이지 코드 개발	setting 코드 개발	메인 메시지 코드 개발	물 주기 설정 코드 개발	
		홈 프래그먼트 DB 연동	마이페이지 DB 연동	setting DB 연동	메인 메시지 DB 연동	물 주기 설정 DB 연동	
최정원	일반 로그인, 소셜 로그인 레이아웃	식물 일지 프래그먼트 레이아웃	식물 일지 작성 액티비티 레이아웃	덧글 액티비티 레이아웃	게시판 프래그먼트 레이아웃	식물 사전 프래그먼트 레이아웃	
	로그인 작동 코드 개발	식물 일지 프래그먼트 코드 개발	식물 일지 작성 액티비티 코드 개발	덧글 액티비티 코드 개발	게시판 프래그먼트 코드 개발	식물 사전 프래그먼트 코드 개발	
	로그인 DB 연동	식물 일지 프래그먼트 DB 연동	식물 일지 작성 액티비티 DB 연동	덧글 액티비티 DB 연동	게시판 프래그먼트 DB 연동	식물 사전 프래그먼트 DB 연동	

## 04. 주요 소스 코드

# 클릭 리스너

## setOnClickListener

```
//MyPageFragment 로 이동
// 상대방의 프로필 이미지를 클릭하면 상대방의 유저 정보로 이동
viewholder.diaryitem_profile_image.setOnClickListener {
    val fragment = MyPageFragment()
    val bundle = Bundle()

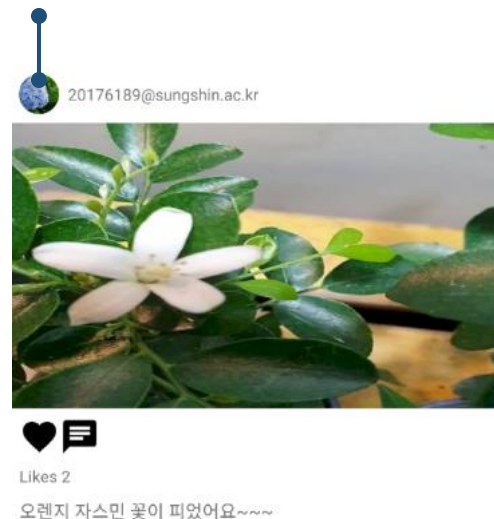
    bundle.putString("destinationUid",
contentDTOs[position].uid) // 이전 페이지에서 넘어온 값
    bundle.putString("userId", contentDTOs[position].userId) //
이메일 값

    fragment.arguments = bundle
    activity?.supportFragmentManager?.beginTransaction()
        ?.replace(R.id.main_content, fragment)?.commit()
}

viewholder.diaryitem_comment_imageview.setOnClickListener { v ->
    var intent = Intent(v.context, CommentActivity::class.java)
    intent.putExtra("contentUid", contentUidList[position])
    startActivity(intent)
}
```

DiaryFragment.kt

프로필 이미지 클릭  
시 유저 정보로 이동



DiaryFragment.kt



MyPageFragment.kt



DiaryFragment.kt

댓글 이미지 클릭 시  
댓글 페이지로 이동



CommentActivity.kt

# 프래그먼트 -> 프래그먼트 이동

Bundle을 사용하여 이동

```

override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position:
Int) {
    var imageview = (holder as CustomViewHolder).imageview

    // Glide로 이미지 다운로드

    Glide.with(holder.itemView.context).load(contentDTOS[position].imageUrl)
        .apply(RequestOptions().centerCrop()).into(imageview)

    holder.imageview.setOnClickListener {

        var myPagePostFragment = MyPagePostFragment()
        val bundle = Bundle()

        bundle.putString("explain", contentDTOS[position].explain)
        bundle.putString("imageUrl", contentDTOS[position].imageUrl)
        bundle.putString("userId", contentDTOS[position].userId)
        bundle.putInt("favoriteCount",
contentDTOS[position].favoriteCount)
        bundle.putString("destinationUid", uid) // 각 유저 페이지의 정보
uid를 보낸다.
        bundle.putLong("timestamp", contentDTOS[position].timestamp!!) //
구분자인 timestamp를 보낸다.
        myPagePostFragment.arguments = bundle

        activity?.supportFragmentManager?.beginTransaction()
            ?.add(R.id.main_content, myPagePostFragment)
            ?.addToBackStack("settingFragment")?.commit()

    }
}

```

MyPageFragment.kt (출발지)

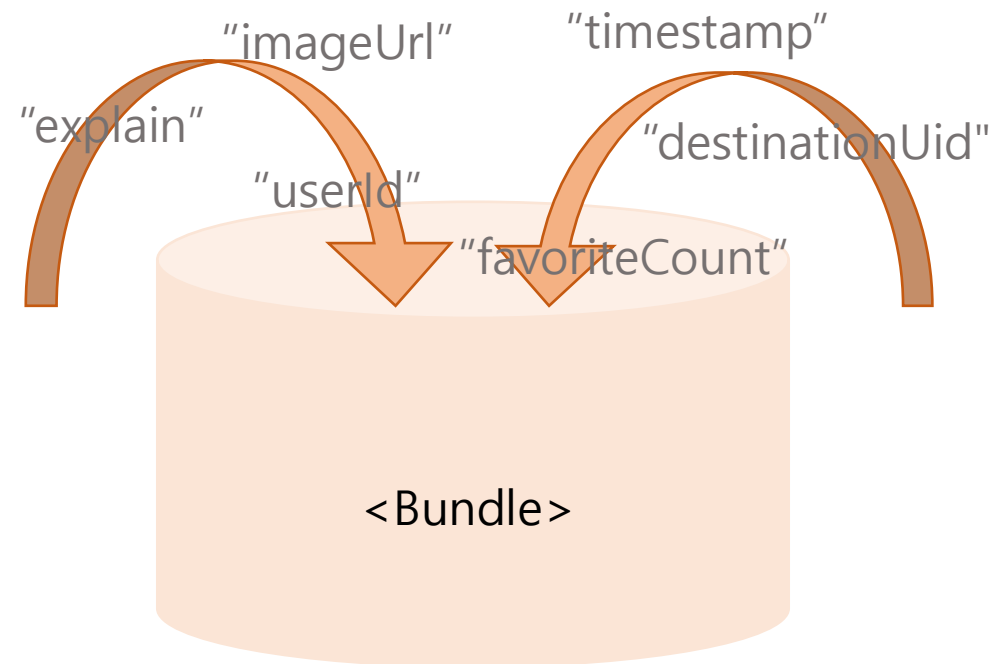
```

// 사용자 게시물 사진 띄우기
Glide.with(activity!!).load(arguments?.getString("imageUrl"))
    .into(view?.my_page_item_imageview_content!!)

```

MyPagePostFragment.kt (도착지)

< Bundle에 값을 넣고 이동 >



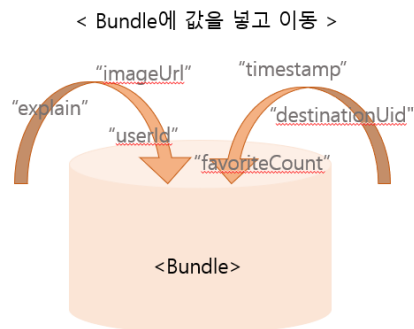
## 프래그먼트 -> 프래그먼트 이동



MyPageFragment.kt (출발지)

```
explain: "페라고늄이라는 허브식물을 데려왔어요!"
favoriteCount: 3
favorites: {NFehYPhwqbdQ0EvUpoOH9QoCZ...}
imageUrl: "https://firebasestorage.googleapis.com/v0/b/complant-67486.appspot.com/o/images%2FIMAGE_20210613_041937-alt=media&token=30cc8233-1aef-4027-9e50-6237b05d6668"
timestamp: 1623525580095
uid: "b00CTv8PIQYvOhB0duzH6ZSwXB62"
userId: "20176189@sungshin.ac.kr"
```

Database (중간 지점)  
Bundle에 있는 key 값을 꺼내어 해당하는 데이터를 도착지에 보낸다.



MyPagePostFragment.kt (도착지)

## 프래그먼트 -> 액티비티 이동

Intent를 사용하여 이동

```
// DetailDictionaryActivity 로 보내줄 데이터들
viewholder.dictionary_cardview.setOnClickListener { v ->
    var intent = Intent(v.context, DetailDictionaryActivity::class.java)
    intent.putExtra("plant_name", plantDictionary!![position].plant_name)
    intent.putExtra("plant_image", plantDictionary!![position].plant_image)
    intent.putExtra("plant_water_cycle",
plantDictionary!![position].plant_water_cycle)
    intent.putExtra("plant_sunshine",
plantDictionary!![position].plant_sunshine)
    intent.putExtra("plant_humidity",
plantDictionary!![position].plant_humidity)
    intent.putExtra("plant_explain",
plantDictionary!![position].plant_explain)
    startActivity(intent)
}
```

DictionaryFragment.kt (출발지)

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_detail_dictionary)
```

```
    plant_name = intent.getStringExtra("plant_name")
    plant_water_cycle =
intent.getStringExtra("plant_water_cycle")
    plant_sunshine = intent.getStringExtra("plant_sunshine")
    plant_humidity = intent.getStringExtra("plant_humidity")
    plant_explain = intent.getStringExtra("plant_explain")
    plant_image = intent.getStringExtra("plant_image")

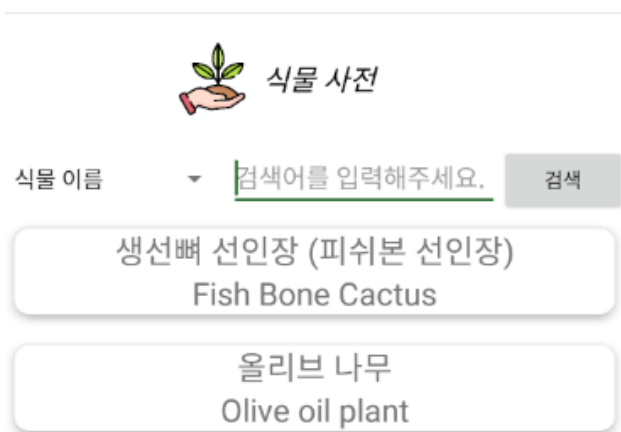
    plant_name_layout.setText(plant_name)
    plant_water_cycle_layout.setText(plant_water_cycle)
    plant_sunshine_layout.setText(plant_sunshine)
    plant_humidity_layout.setText(plant_humidity)
    plant_explain_layout.setText(plant_explain)

    Glide.with(this).load(plant_image).into(plant_image_layout)
}
```

DetailDictionaryActivity.kt (도착지)



# 프래그먼트 -> 액티비티 이동



DictionaryFragment.kt (출발지)

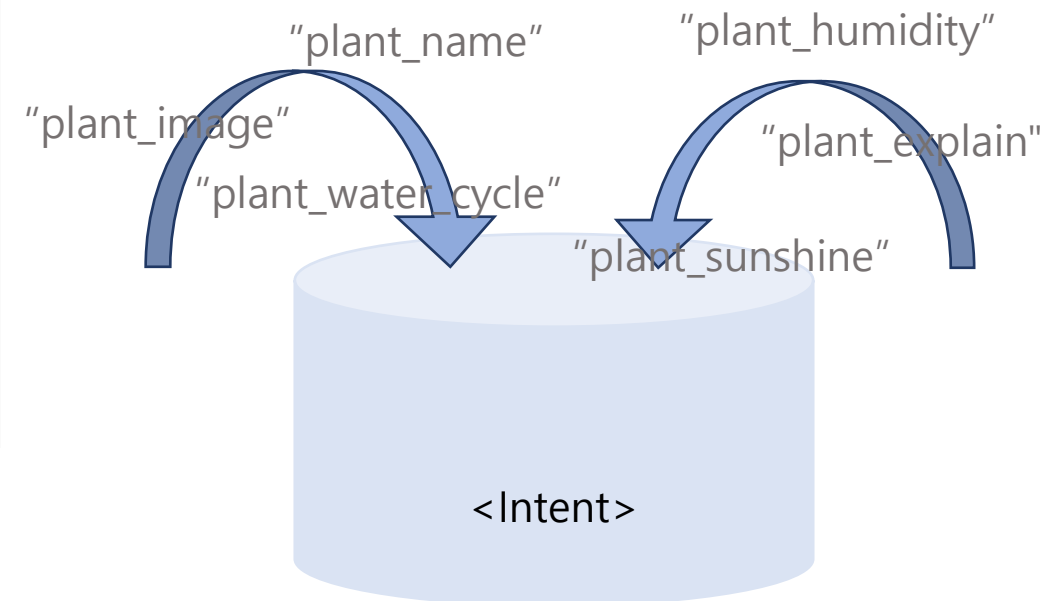


DetailDictionaryActivity.kt (도착지)

```
plant_eng_name: "Olive oil plant"
plant_explain: "역사가 가장 오래된 과일나무 중의 하나인 올리브 나무는 생명력이 굉장히 강한 식물이에요. 연중 일교차가 적고 강수량도 적은 지중해에서 많이 재배되고 있어요. 강렬한 햇볕이 있다면 키우는 데 큰 무리가 없답니다. 게다가 화분에서도 열매를 맺을 수 있는 몇 안 되는 식물이라 인기가 많아요. 열매인 올리브는 몸속 유해 산소를 차단하는 강력한 항산화 작용이 있어서 오일 이외에도 다양한 요리에 쓰이는 중요한 식자재랍니다."
plant_humidity: "[40~70%] 보통날의 환경에 적합해요."
plant_image: "data:image/jpeg;base64,/9j/4AAQSkZJRgABAQAAQAB...
plant_name: "올리브 나무"
plant_sunshine: "[양지] 햇빛이 있는 곳에서 잘자라요."
plant_water_cycle: "[주 1~2회] 흙 표면부터 3cm까지 마르면, 듬뿍 주세요."
```

Database (중간 지점)  
Intent에 있는 key 값을 꺼내어 해당하는 데이터를 도착지에 보낸다.

< Intent에 값을 넣고 이동 >



# Glide

```
fun getProfileImage() {
    firestore?.collection("profileImages")?.document(previousUid!!)
        ?.addSnapshotListener { documentSnapshot,
            firebaseFirestoreException ->
```

```
        // documentSnapshot 이 null 일 경우 바로 리턴
        if (documentSnapshot == null) return@addSnapshotListener

        // null 이 아닐 경우 이미지 주소 값을 받아와서 Glide로 다운로드
        받는다.
```

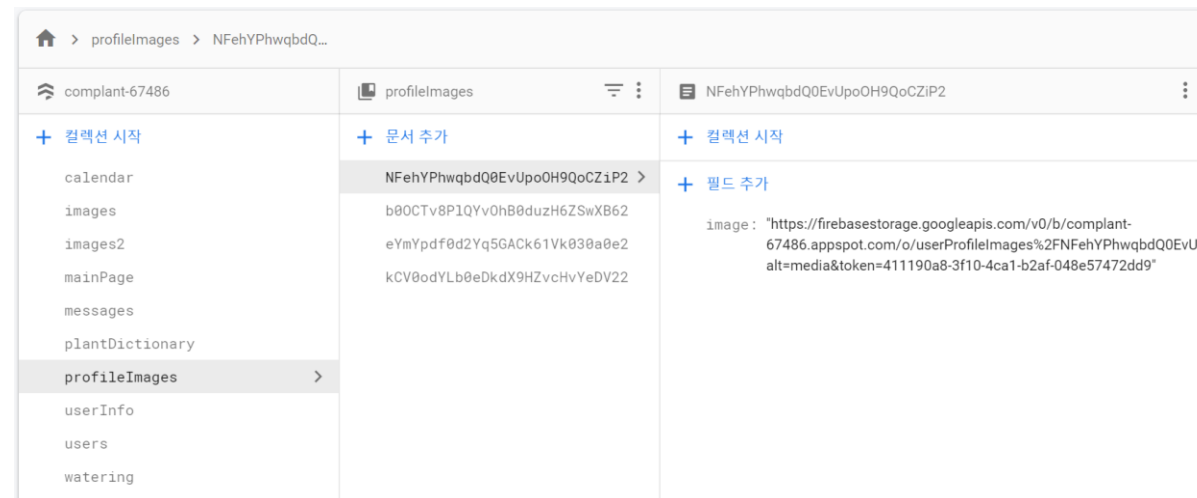
```
        if (documentSnapshot.data != null) {
            var url = documentSnapshot?.data!!["image"]
```

```
Glide.with(activity!!).load(url).apply(RequestOptions().circleCrop())
    .into(view?.my_page_diaryitem_profile_image!!)
```

```
        }
    }
}
```

## HomeFragment.kt

```
// 사용자 프로필 사진 띄우기
getProfileImage()
```



업로드한 이미지를 DB에서 가져온다.



실행 결과

HomeFragment.kt

## 데이터 조회

## addSnapshotListener 이용

```

firestore?.collection("mainPage")?.document(uid!!)?.addSnapshotListener
{ snapshot, e ->
    if (snapshot == null) return@addSnapshotListener
    var mainPageDTO = snapshot.toObject(MainPageDTO::class.java)
    if (mainPageDTO?.message != null) {
        view.speech_box_text.setText(mainPageDTO?.message)
    }
}

```

## HomeFragment.kt

complant-67486	mainPage	b00CTv8PIQYvOhB0duzH6ZSwXB62
+ 컬렉션 시작	+ 문서 추가	+ 컬렉션 시작
calendar	NFehYPhwqbdQ0EvUpo0H9QoCZiP2	
images	b00CTv8PIQYvOhB0duzH6ZSwXB62 >	+ 필드 추가
images2	eYmYpdf0d2Yq5GACk61Vk030a0e2	message: "오늘 싱글이 물 주는거 잊지 말자!"
mainPage >	kCV0odYLB0eDkdX9HZvcHvYeDV22	uid: "b00CTv8PIQYvOhB0duzH6ZSwXB62"
messages		
plantDictionary		
profileImages		
userInfo		
users		
watering		



오늘 싱글이 물 주는거 잊지 말자!

싱글이♡  
(수국)



wonny 님의 싱글이♡이(가)  
쑥쑥 자라고 있어요!



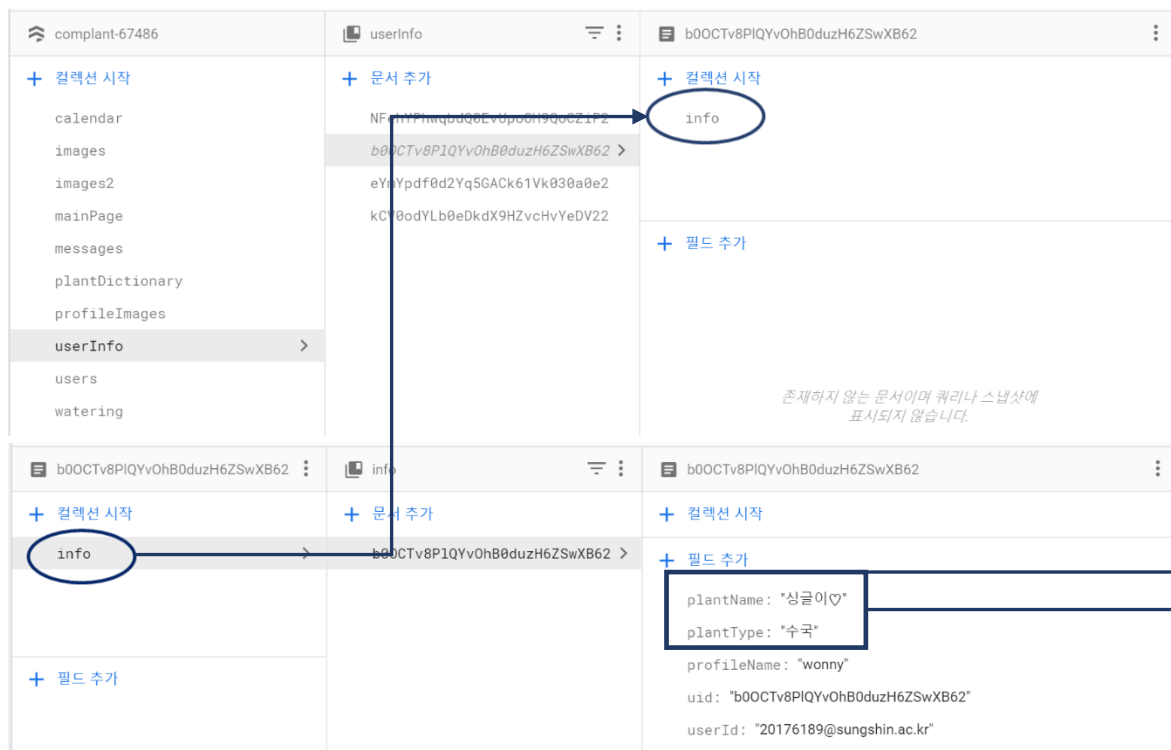
HomeFragment.kt

# 데이터 조회

## addSnapshotListener 이용

```
// DB에서 profile name과 plant name을 가져와서 현재 정보를 보여준다.
firestore?.collection("userInfo")
?.document(uid!!)
?.collection("info")
?.document(uid!!)
?.addSnapshotListener { snapshot, e ->
    if (snapshot == null) return@addSnapshotListener
    var userInfoDTO =
        snapshot.toObject(UserInfoDTO.UserInfo::class.java)
    view.profile_name.setText(userInfoDTO?.profileName)
    view.plant_name.setText(userInfoDTO?.plantName)
    view.plant_name2.setText(userInfoDTO?.plantName)
    view.plant_type.setText("(" + userInfoDTO?.plantType + ")")
}
```

### HomeFragment.kt



오늘 싱글이 물 주는거 잊지 말자!

싱글이♡  
(수국)



wonny님의 싱글이♡이(가)  
복복 자라고 있었어요!



HomeFragment.kt

# 데이터 수정

## Set 함수 사용

```
var userInfoDTOs = UserInfoDTO.UserInfo()
```

// 완료 버튼을 누르면 정보 DB에 업데이트

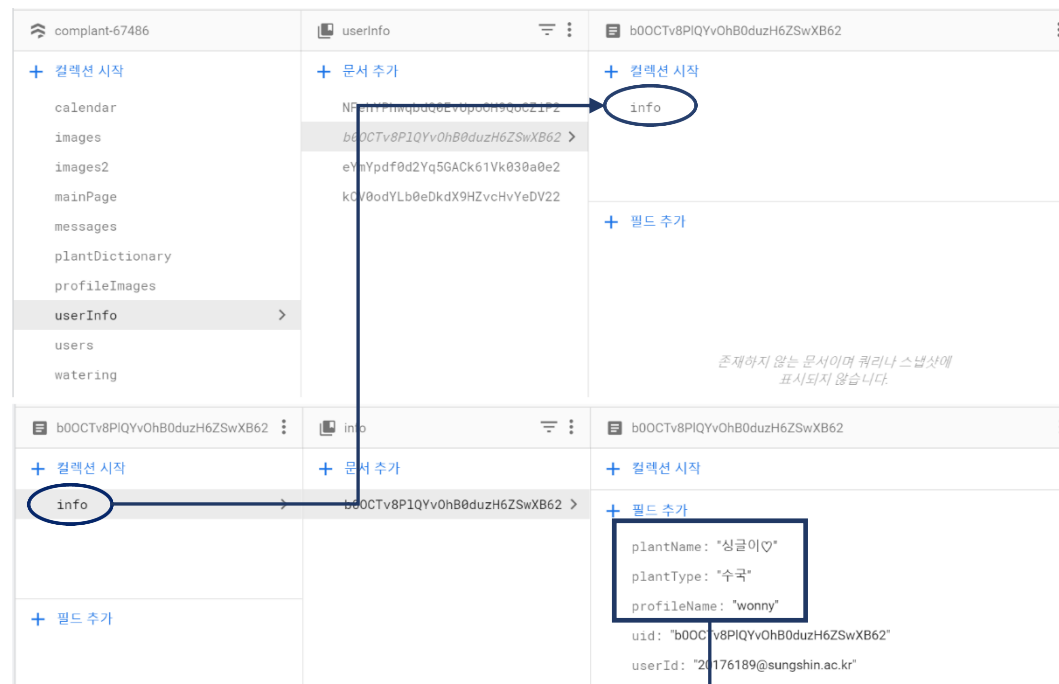
```
view.btn_my_info_update.setOnClickListener {
    userInfoDTOs?.profileName = edit_profile_name_setting.text.toString()
    userInfoDTOs?.plantName = edit_plant_name_setting.text.toString()
    userInfoDTOs?.plantType = edit_plant_type_setting.text.toString()
    userInfoDTOs?.uid = uid
    userInfoDTOs?.userId = currentUserId
}
```

```
firestore?.collection("userInfo")?.document(uid!!)?.collection("info")?.document(uid!!)?.set(userInfoDTOs)
```

```
mainActivity?.goBack()
```

```
}
```

myInformationFragment.kt



## 내 정보 변경



ID : 20176189@sungshin.ac.kr  
 Profile name : wonny  
 Plant name : 싱글이♡  
 Plant type : 수국

Profile name	wonny
Plant name	싱글이♡
Plant Type	수국

완료

myInformationFragment.kt

# 리사이클러 뷰

```

inner class CalendarRecyclerViewAdaper :
RecyclerView.Adapter<RecyclerView.ViewHolder>() {
    // CalendarDTO 클래스 ArrayList 생성
    var calendarDTOs: ArrayList<CalendarDTO.Watered> = arrayListOf()
    var stringDate: String? = null

    init {
        firestore?.collection("calendar")
            ?.document(uid!!)
            ?.collection("userWatered")
            ?.orderBy("wateredString", Query.Direction.DESENDING)
            ?.addSnapshotListener { querySnapshot,
                firebaseFirestoreException ->
                // 배열 비움
                calendarDTOs.clear()

                // querySnapshot이 null 일 경우, 바로 종료시킨다.
                if (querySnapshot == null) return@addSnapshotListener

                for (snapshot in querySnapshot!!.documents) {
                    var item =
                        snapshot.toObject(CalendarDTO.Watered::class.java)
                    calendarDTOs.add(item!!)
                }
                notifyDataSetChanged()
            }
    }
}

```

CalendarFragment.kt

```

// xml 파일을 inflate하여 ViewHolder를 생성
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RecyclerView.ViewHolder {
    var view =
        LayoutInflater.from(parent.context).inflate(R.layout.item_calendar, parent, false)
    return CustomViewHolder(view)
}

inner class CustomViewHolder(view: View) : RecyclerView.ViewHolder(view)

override fun getItemCount(): Int {
    return calendarDTOs.size
}

override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {
    var viewholder = (holder as CustomViewHolder).itemView

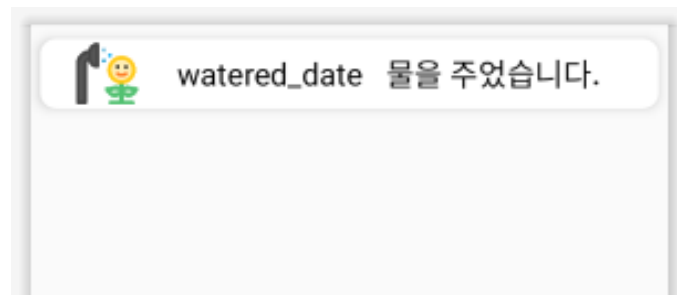
    if (calendarDTOs!![position].wateredMonth!! < 10 && calendarDTOs!![position].wateredDay!!
    < 10) {
        stringDate =
            calendarDTOs!![position].wateredYear.toString() + ".0" +
            calendarDTOs!![position].wateredMonth.toString() + ".0" +
            calendarDTOs!![position].wateredDay.toString();
    } else if (calendarDTOs!![position].wateredMonth!! < 10) {
        stringDate =
            calendarDTOs!![position].wateredYear.toString() + ".0" +
            calendarDTOs!![position].wateredMonth.toString() + "." +
            calendarDTOs!![position].wateredDay.toString();
    } else if (calendarDTOs!![position].wateredDay!! < 10) {
        stringDate =
            calendarDTOs!![position].wateredYear.toString() + "." +
            calendarDTOs!![position].wateredMonth.toString() + ".0" +
            calendarDTOs!![position].wateredDay.toString();
    } else {
        stringDate =
            calendarDTOs!![position].wateredYear.toString() + "." +
            calendarDTOs!![position].wateredMonth.toString() + "." +
            calendarDTOs!![position].wateredDay.toString();
    }

    viewholder.watered_date.text = stringDate
}
}

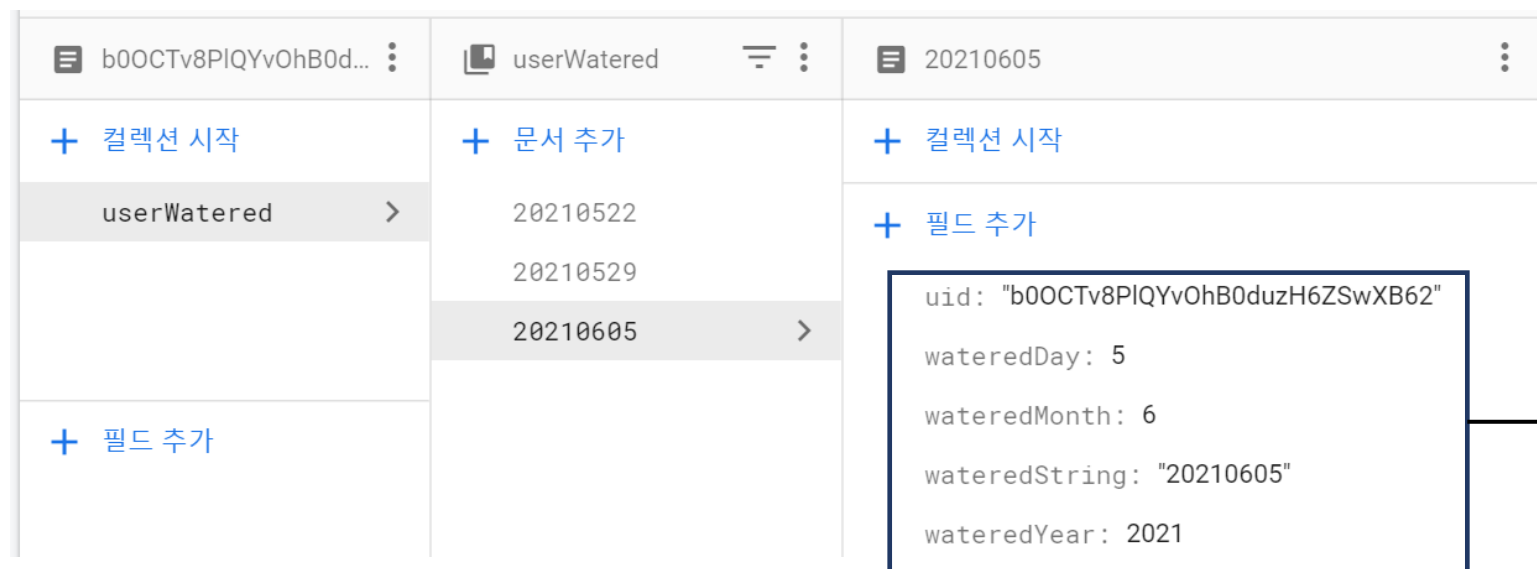
```



# 리사이클러 뷰



Item\_calendar.kt



CalendarFragment.kt

## 05. 향후 계획 및 느낀 점



---

## 향후 계획

- home fragment의 D+Day(식물의 성장 날짜 기록) 추가
- 물 주는 날 예정일 표시 추가
- 중복 확인(사용자 프로필 이름)
- MyPagePost 좋아요, 댓글 정보 받아오기
- 회원 탈퇴
- 게시글, 메시지 리스트, 물 주기에 대한 '삭제' 기능 추가
- Play 스토어 애플리케이션 등록

---

## 느낀 점

- Kotlin이라는 새로운 언어를 배우면서 프로젝트를 진행하게 되어 더 도전적이고, 재미있었다.
- 프로젝트를 직접 만들어 보며 배우는 것들이 이론 공부를 하며 얻는 것들보다 더 효과적임을 알게 되었다.
- 프로젝트의 전반적인 진행 단계에 대해 익힐 수 있었다.

## 06. 실행 화면 동영상



감사합니다~😊