

셔틀 버스 이용시 안전 사고 예방을 위한

하차 시스템 어플리케이션 셔테이션

양경민

email:mi568977@naver.com

하차 시스템 어플리케이션 - 서태이션

01 프로젝트 개요

02 어플리케이션 구성 및 DB 설계

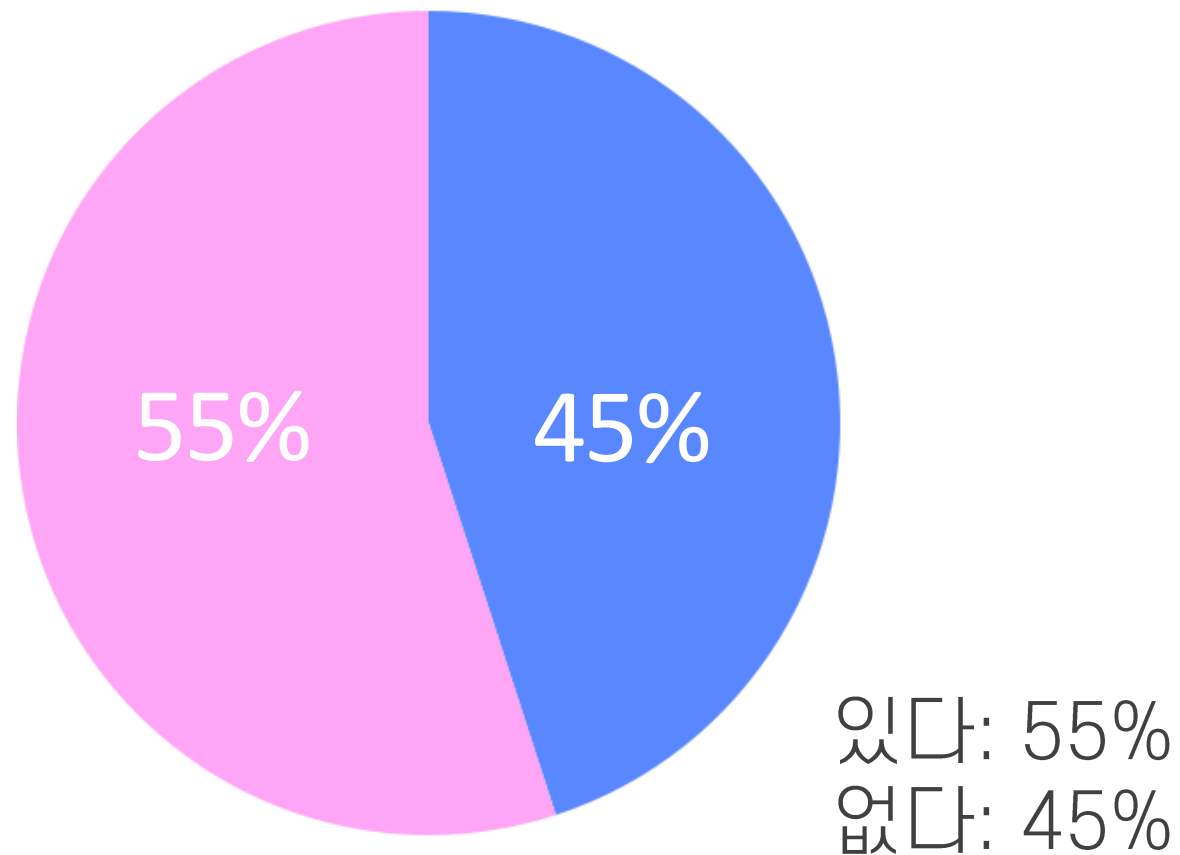
03 기능 구현

04 기대효과

01 프로젝트 개요

셔틀 버스 안전사고 위험성 조사

셔틀버스 안전 사고발생에 대해서 생각해본 적 있나요?



‘있다’라고 대답한 55%

이유는 무엇인가요?

- 셔틀버스 내 하차 벨의 부재로 미리 서있어야 한다.
- 하차 시에 운전기사가 하차 인원을 모르기 때문에 급하게 하차해야 하는 경우가 있다.
- 다리를 다쳤을 때 셔틀버스가 완전히 정차하면 내리려고 기다리고 있었는데 정류장을 정차하지 않고 지나간 경험이다.

- 교내 셔틀버스 이용자 500명 대상

01 프로젝트 개요

주요 기능

- 셔틀버스 이용자에게 버스이용시 필요한 편의 기능 제공
- 스마트폰을 이용하여 원하는 하차 지역선택
- 안전하고 편리한 하차 활성화

제작 의도

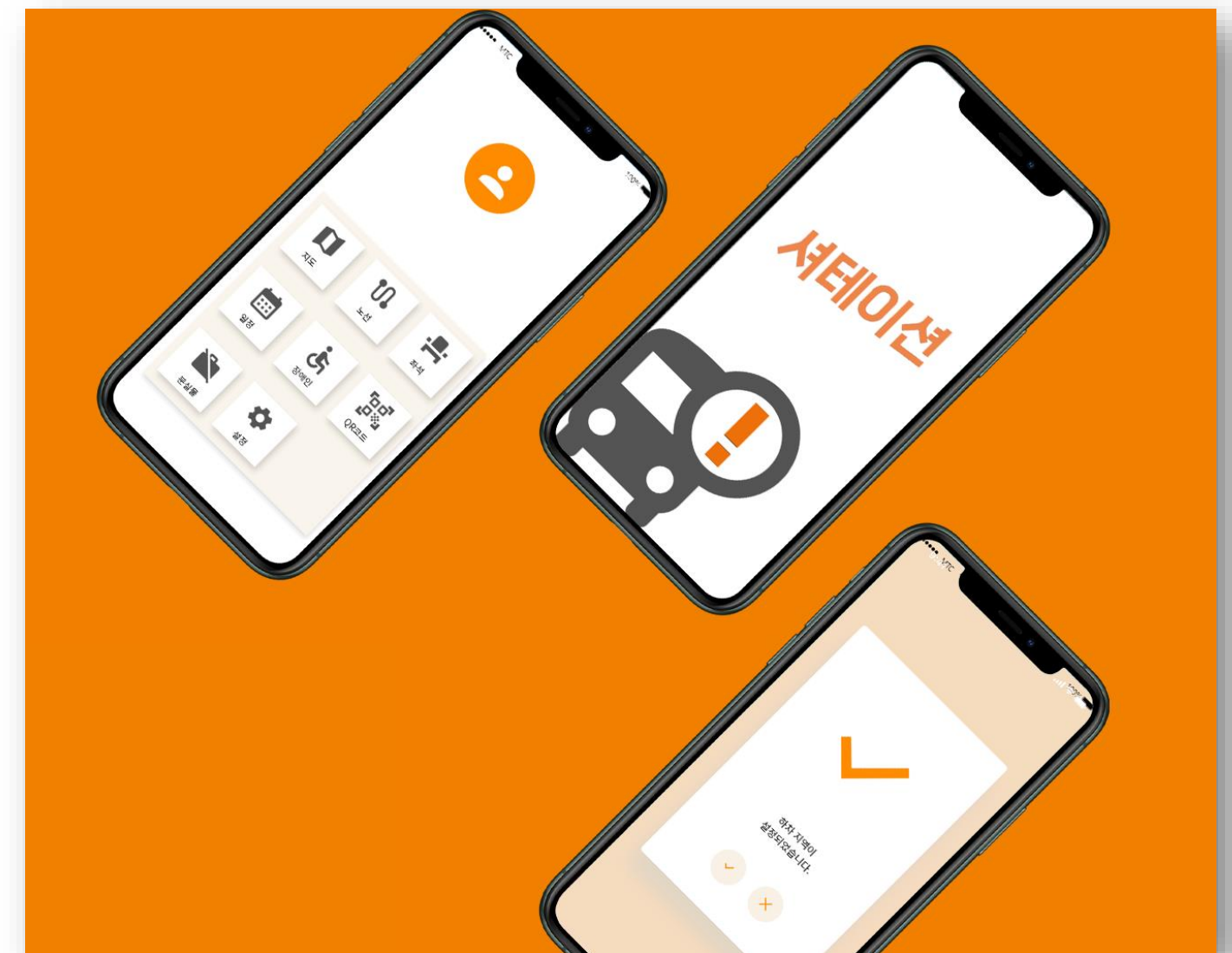
- ① 하차 시에 발생하는 안전사고의 위험성 증가
 - ② 하차 벨의 부재로 발생하는 정차하지 않고 정류장을 지나가는 상황 발생
- ✓ 하차 시스템 어플리케이션 '셔테이션' 제작!

- 개발 인원:3(명)
- 개발 기간: 2022.09.05~2022.12.3
- 맡은 역할:
 - (1)회원가입 및 로그인 기능
 - (2)하차 신청 기능 구현
 - (3)회원 데이터 베이스 구축
 - (4)하차 신청 데이터 베이스 구축

01 프로젝트 개요

셔테이션

- ‘셔틀버스’와 ‘역’을 의미하는 스테이션의 합성어



01 프로젝트 개요

개발 환경 및 언어



Android
Studio



Kotlin

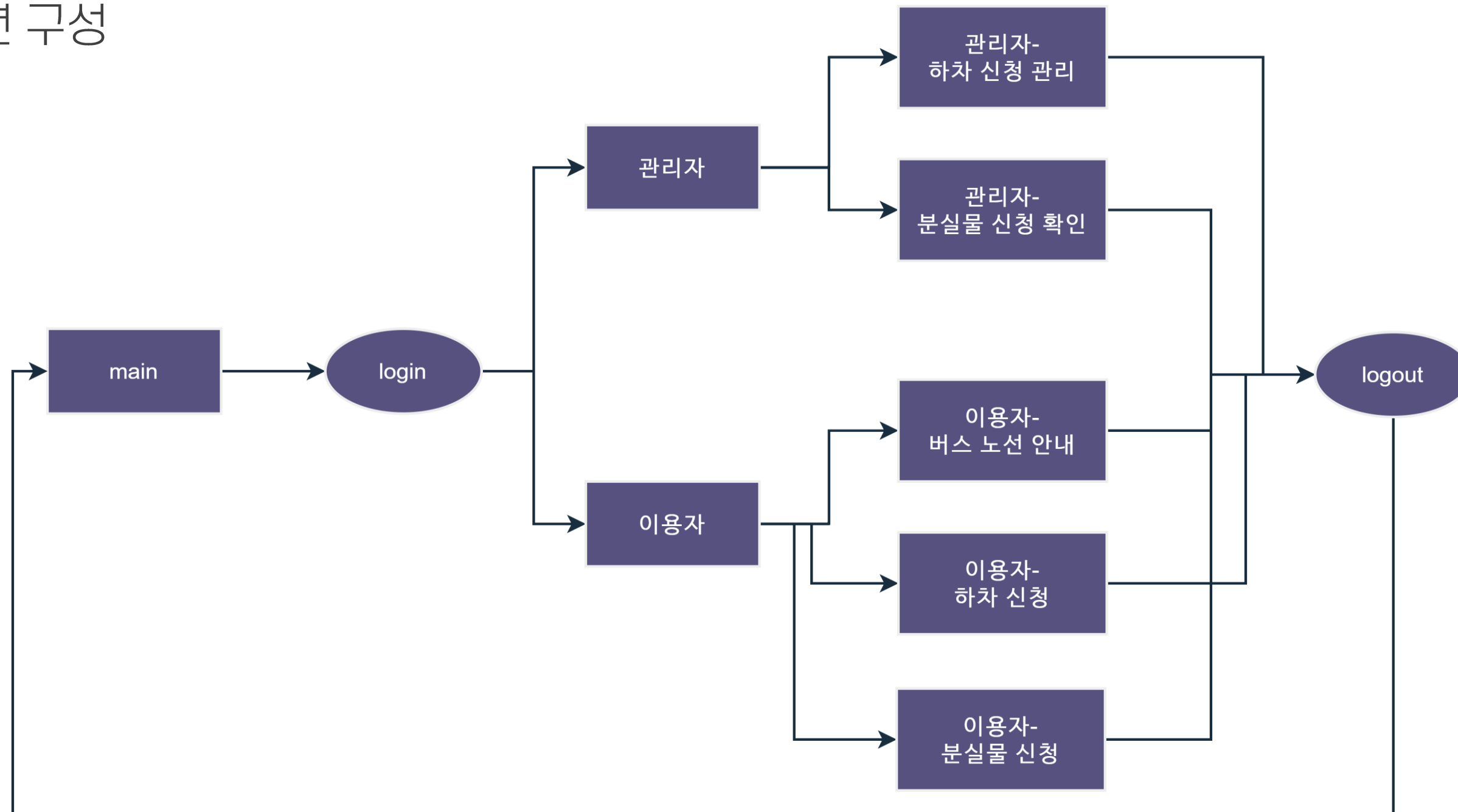


Firebase



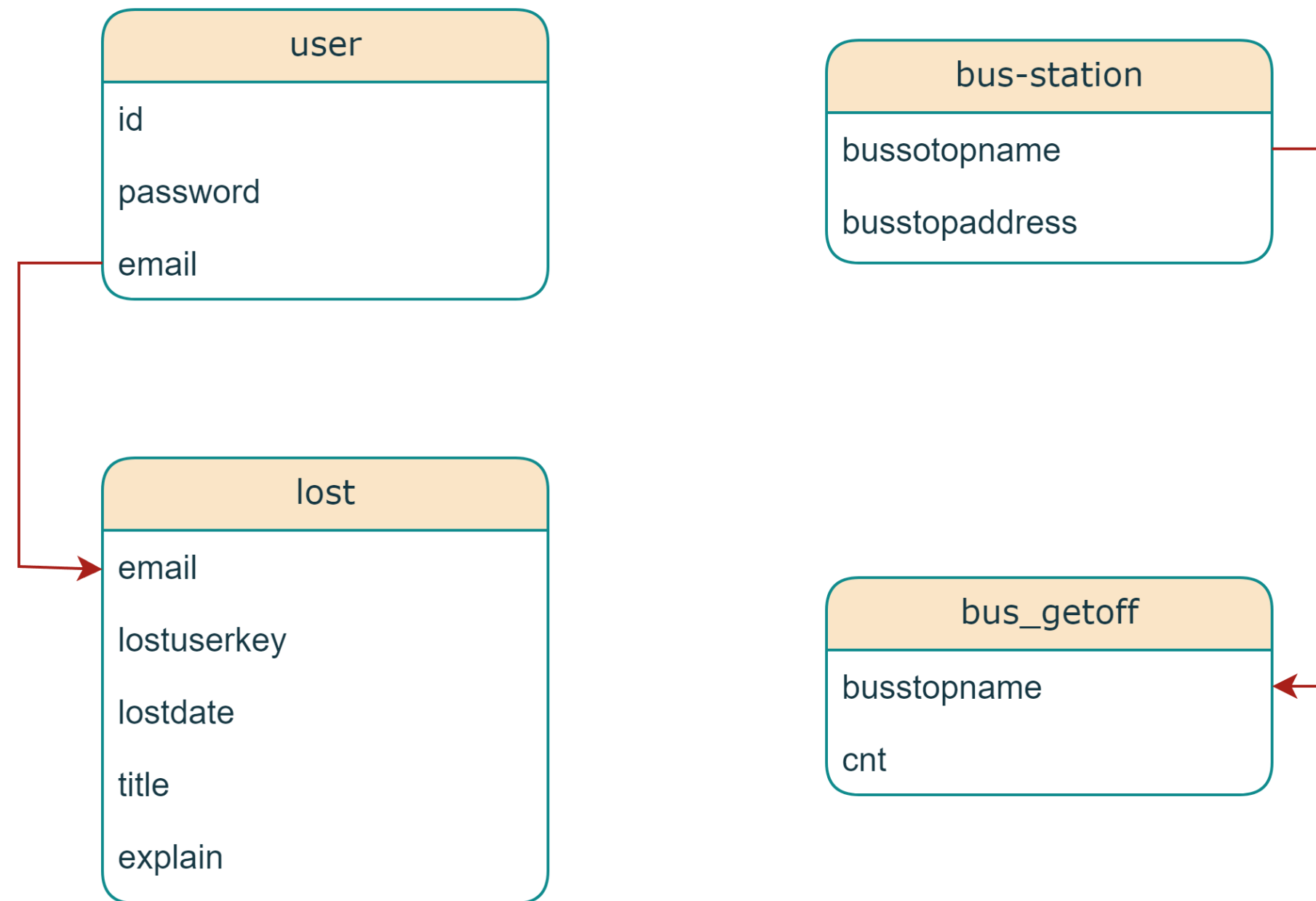
02 어플리케이션 구성 및 DB 설계

어플리케이션 구성

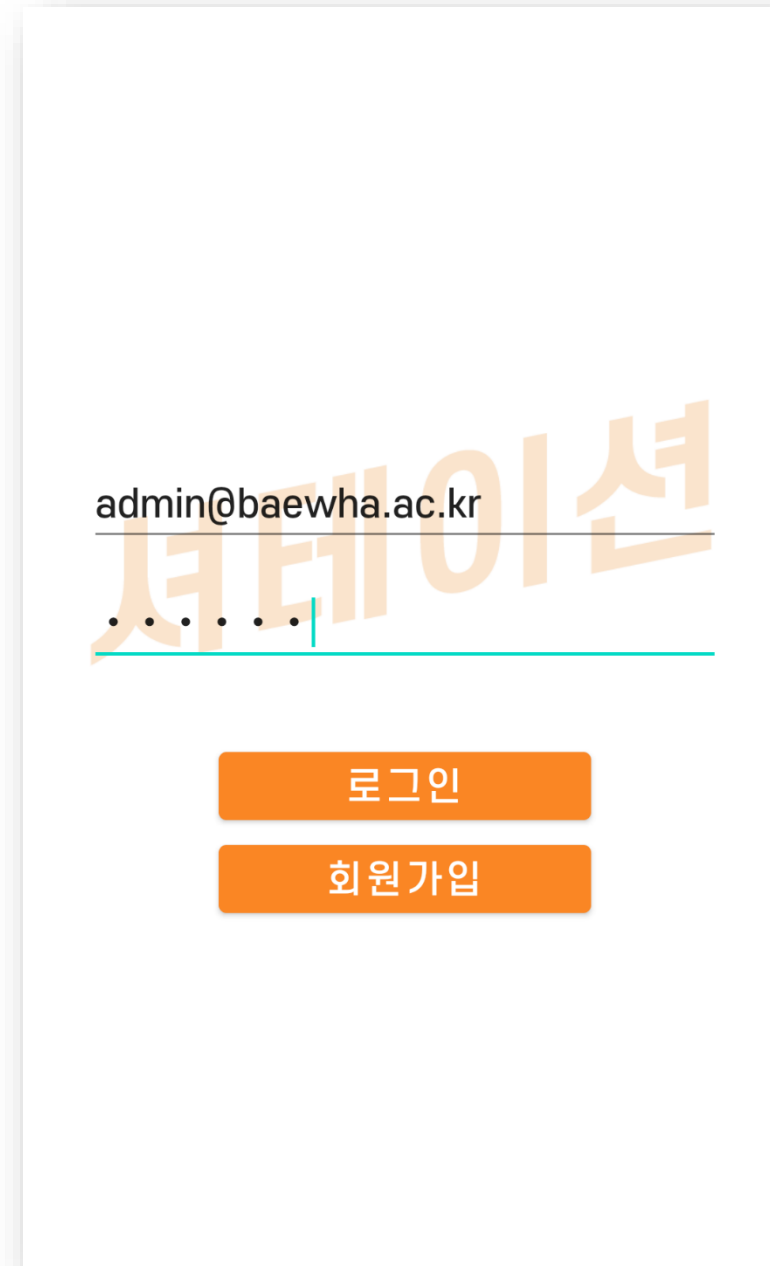
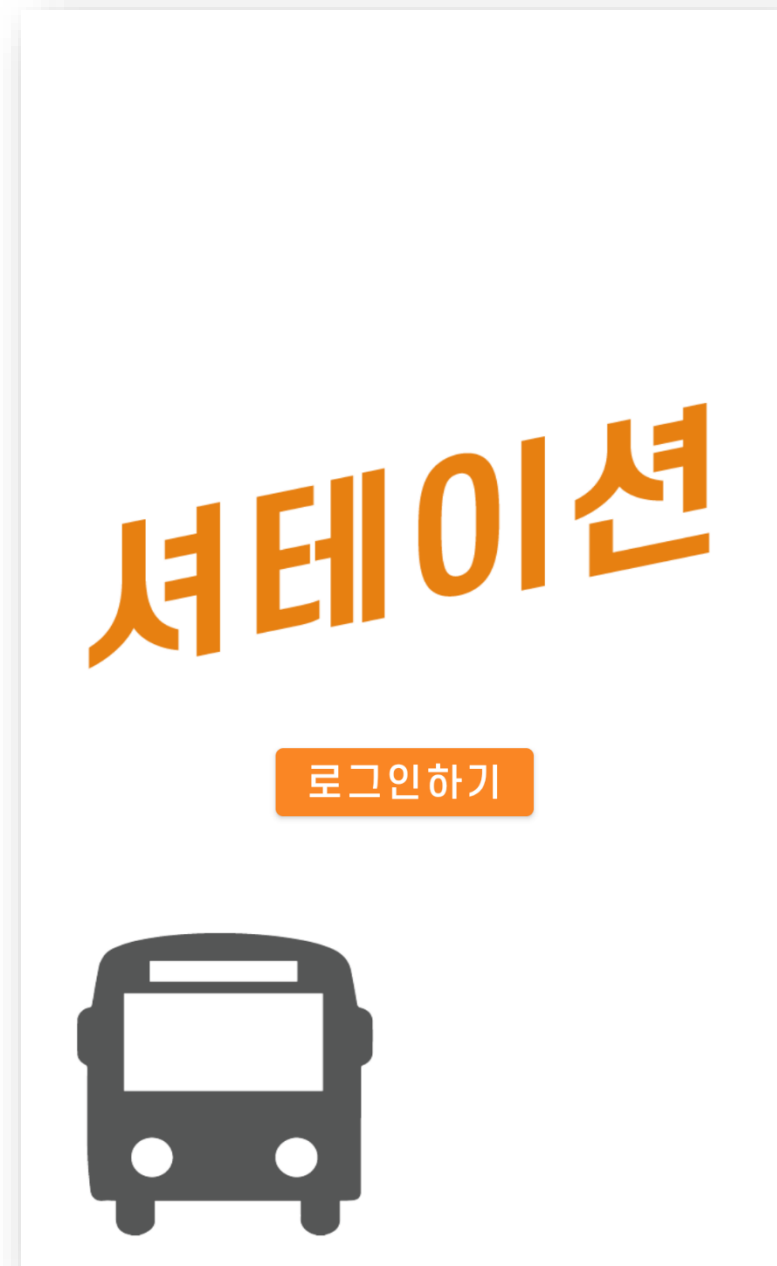


02 어플리케이션 구성 및 DB 설계

DB 설계



03 기능 구현



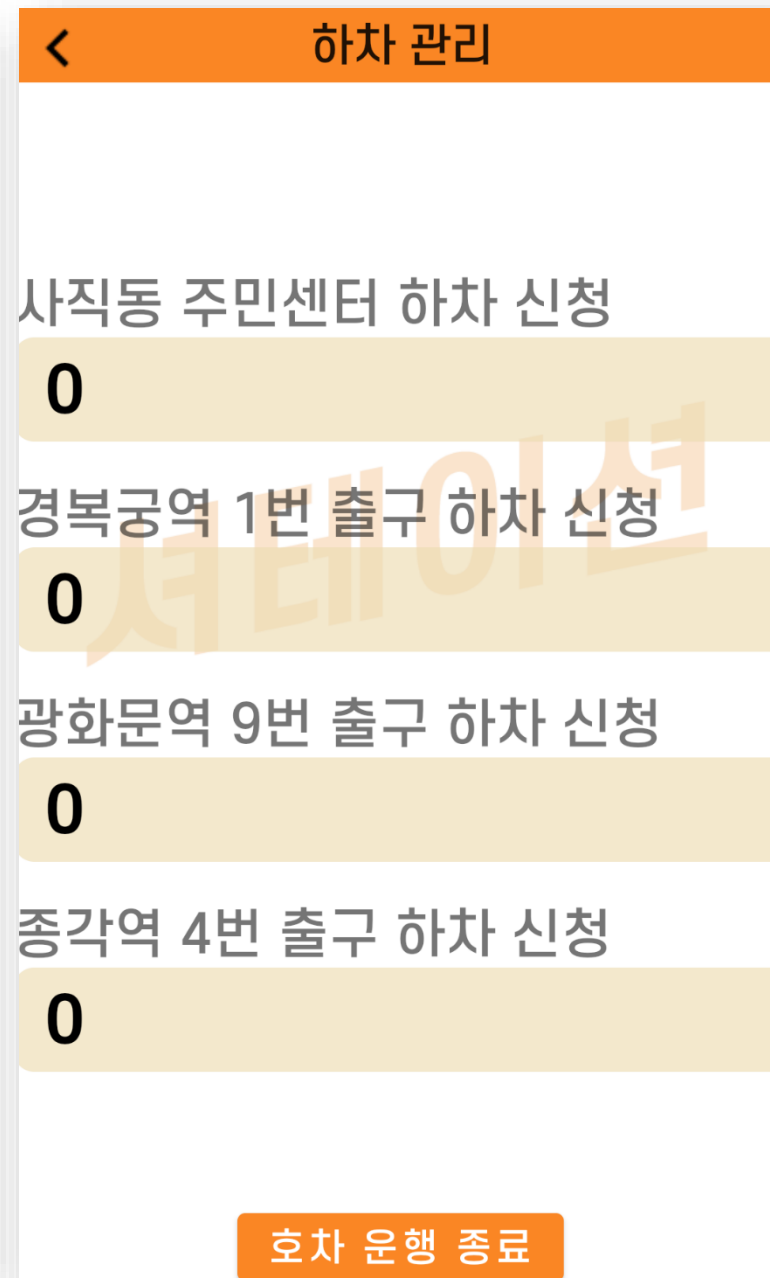
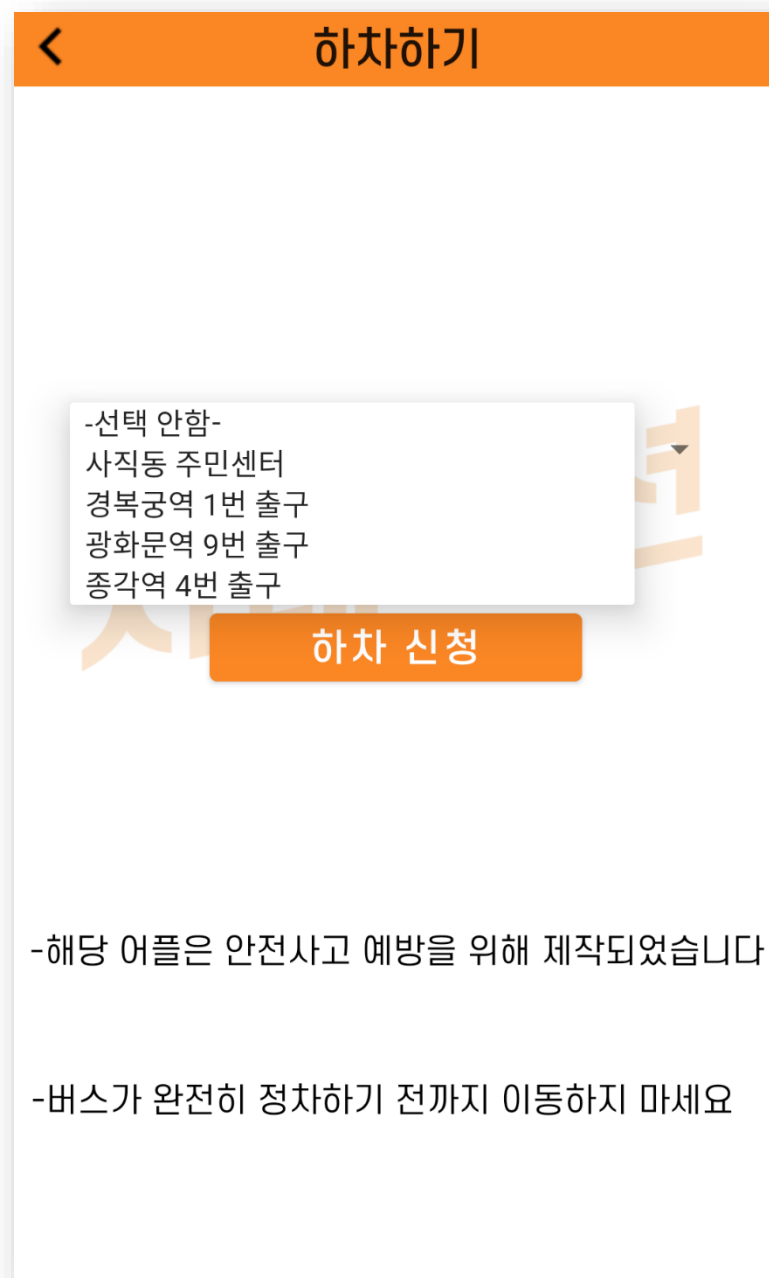
```
class RegisterActivity : AppCompatActivity() {  
    val binding by lazy{ ActivityRegisterBinding.inflate(layoutInflater) }  
    private var auth : FirebaseAuth = FirebaseAuth.  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(binding.root)  
  
        auth= FirebaseAuth  
        binding.registerend.setOnClickListener { it: View!  
            initSignup()  
        }  
    }  
    private fun initSignup() { //성공!!  
        binding.registerend.setOnClickListener { it: View!  
            val email = binding.registerId.text.toString()  
            val password = binding.registerPw.text.toString()  
  
            auth.createUserWithEmailAndPassword(email,password)  
                .addOnCompleteListener { task ->  
                    if(task.isSuccessful){  
                        Toast.makeText( context: this, text: "회원가입에 성공했습니다!",Toast.LENGTH_SHORT).show()  
                        val intent= Intent( packageContext: this,LoginActivity::class.java)  
                        startActivity(intent)  
                    }else{  
                        Toast.makeText( context: this, text: "이미 존재하는 계정이거나, 회원가입에 실패했습니다.",Toast.LENGTH_SHORT).show()  
                    }  
                }  
        }  
    }  
}
```

03 기능 구현



```
binding.adbus.setOnClickListener{  
    val intent= Intent(this,AdminMapActivity::class.java)//정류장  
    startActivity(intent)  
}  
binding.adoff.setOnClickListener {  
    val intent= Intent(this,AdminGetOffActivity::class.java)//하차 관리  
    startActivity(intent)  
}  
binding.adlost.setOnClickListener {  
    val intent= Intent(this,AdminLostActivity::class.java)//분실물 확인  
    startActivity(intent)  
}  
binding.adcal.setOnClickListener {  
    val intent= Intent(this,CalendarActivity::class.java)//일정 관리  
    startActivity(intent)  
}  
binding.adlogout.setOnClickListener {  
    val intent= Intent(this,MainActivity::class.java)//로그아웃  
    startActivity(intent)  
}
```

03 기능 구현



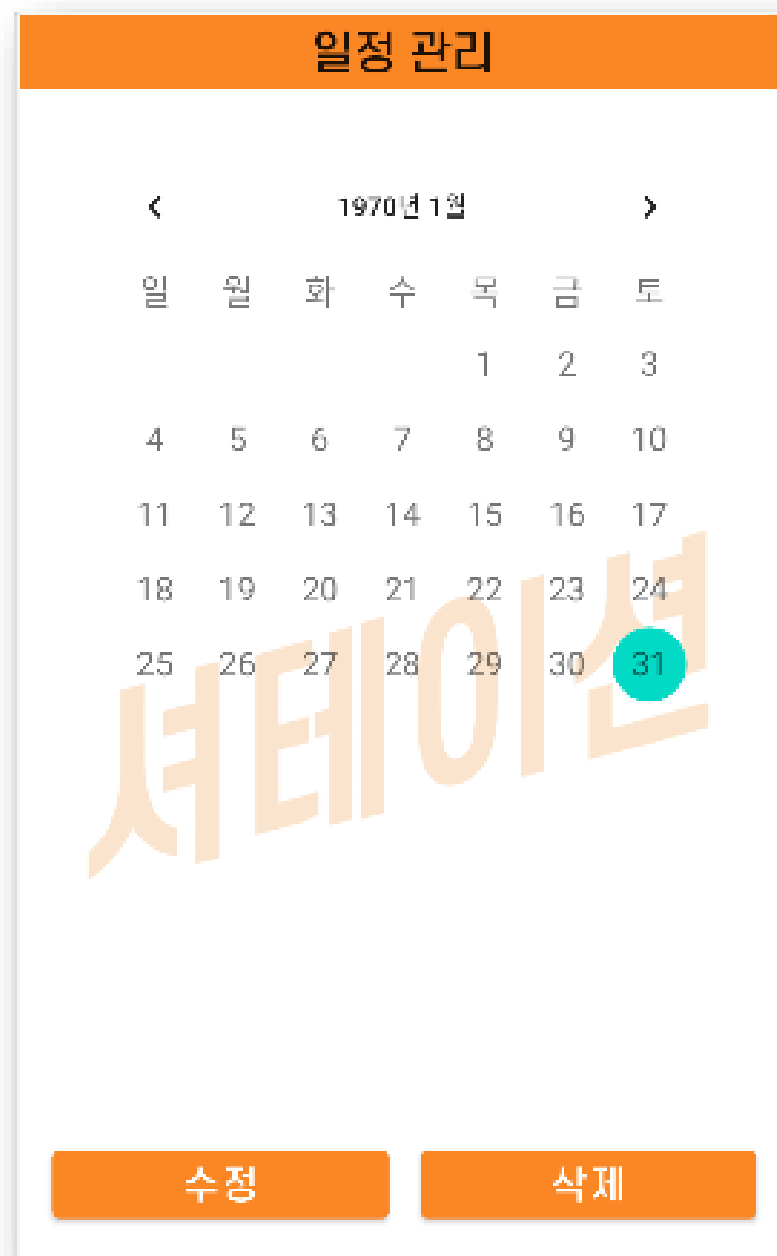
```
var busstopname=binding.offList.selectedItem.toString()
if(busstopname=="사직동 주민센터"){
    binding.resultText.text = busstopname+"에서 하차신청이 되었습니다"
    //goR.off1Count="True"
    goR.offcount="True"
    goR.busname=busstopname
    goR.userId=name
    FirebaseFirestore.getInstance().collection( collectionPath: "busstop1").document(name.toString()).set(goR)

binding.btnReset.setOnClickListener { it: View!
    for(i in 0..goRs1.size-1 step(1)){
        firestore?.collection( collectionPath: "busstop1")?.document(goRs1[i].userId.toString())?.delete()
    }

    firestore?.collection( collectionPath: "busstop1")
        ?.addSnapshotListener { querySnapshot, firebaseFirestoreException ->
            if (querySnapshot == null) return@addSnapshotListener

            for (snapshot in querySnapshot.documents)
                goRs1.add(snapshot.toObject(GoR::class.java)!!)
            binding.off1.text = goRs1.size.toString()
        }
}
```

03 기능 구현



// 달력 내용 조회, 수정

```
fun checkDay(cYear: Int, cMonth: Int, cDay: Int, userID: String) {
```

// 저장할 파일 이름설정

```
fname = "" + userID + cYear + "-" + (cMonth + 1) + "-" + cDay + ".txt"
```

```
var fileInputStream: FileInputStream
```

```
try {
```

```
fileInputStream = openFileInput(fname)
```

```
val fileDate = ByteArray(fileInputStream.available())
```

```
fileInputStream.read(fileDate)
```

```
fileInputStream.close()
```

```
str = String(fileDate)
```

03 기능 구현

<

분실물

틴트

9시 10분차량 분실물 신청합니다.

2022.11.2

202031002@baewha.ac.kr

에어팟pro

아침 8시 50분 분실물 신청합니다

2022.11.3

202031008@baewha.ac.kr

우산

검은색 장우산 분실물 신청합니다

2022.11.9

202031008@baewha.ac.kr

텀블러

스타벅스 투명 텀블러 분실물 신청합니다

2022.11.14

202031009@baewha.ac.kr

보조배터리

라이언 보조배터리 분실물 신청합니다

2022.11.17

201831002@baewha.ac.kr

후드집업

+

<

분실물 신청

분실 날짜

0000.00.00



제목

분실물 제목

설명

분실물 내용

취소

등록하기

```
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(binding.root)
    var auth : FirebaseAuth? = null

    binding.btnlostreg.setOnClickListener {
        val intent= Intent(this,LostregisActivity::class.java)
        startActivity(intent)
    }

    lostList= ArrayList()
    dao= UserDao()
    adapter= LostAdapter(this, lostList)
    binding.recyclerview.layoutManager=LinearLayoutManager(this)
    binding.recyclerview.adapter=adapter
    //사용자 정보 가져오기
    getLostList()
}

private fun getLostList() {
    dao.getLostList()?.addValueEventListener(object: ValueEventListener {
        override fun onDataChange(snapshot: DataSnapshot) {
            for(dataSnapshot in snapshot.children){
                val lostuser=dataSnapshot.getValue(Lostuser::class.java)
                val key=dataSnapshot.key
                lostuser?.lostuserKey=key.toString()
                if(lostuser !=null){
                    lostList.add(lostuser)
                }
            }
            adapter.notifyDataSetChanged()
        }
    })

    override fun onCancelled(error: DatabaseError) {

    }

}
```

04 기대 효과

버스 운행중의 이동을 최소화하여 안전사고 예방



버스 무정차 방지를 통한 셔틀버스 이용 만족도 증가