2024 경동대학교

최종 발표

건강한 흡연문화 조성을 위한 관리분석 및 공간찾기 어플개발

담당자 : 9팀 최경채 , 조성민 ,김상현 | 010-2495-5667 | 2121058@kdu.ac.kr | https://www.kduniv.ac.kr/kor/Main.do

01 설계한 동기 및 목표

흡연자들이 주변에 흡연구역을 찾지못하여 흡연을 하지 못하거나 그로 인해 길거리에서 흡연을 하는것을 목격하고 건강한 흡연문화 생활을 위하여 앱을 설계하게 되었습니다 최종적으로는 흡연자들이 흡연 구역을 쉽게 찾아내고 찾은 구역에 대한 의견을 공유하는 것을 구현하는것이 목표입니다.

02

설계한 요구사항

흡연자들커뮤니티

댓글 기능을 활성화해 흡연자들끼리 정보를 공유 할 수 있게 합니다.

흡연관리

흡연피는 시간 , 흡연 했던 날짜를 통해 관리 할 수 있도록 합니다.

흡연구역찾기

주변에있는 흡연구역을 찾을수 있도록 도와줍니다

03

개발 환경

흡연 어플리케이션

01 안드로이드 스튜디오

자바 기반의 코틀린은 코딩이 좀 더 간단한고 편리하게 수정이가능하여 오류검출에 용이 하기 때문에 사용하였습니다.

02 파이어베이스

사용자 로그인, 흡연구역 과 같은 데이터정보 들을 관리를 하기위해 데이터베이스 시스템인 파이어베이스를 사용하였습니다.

03 네이버 지도

API를 받아와 어플리케이션에 표시하는것은 네이버지도 API를 사용하였습니다

04 공공기관 api

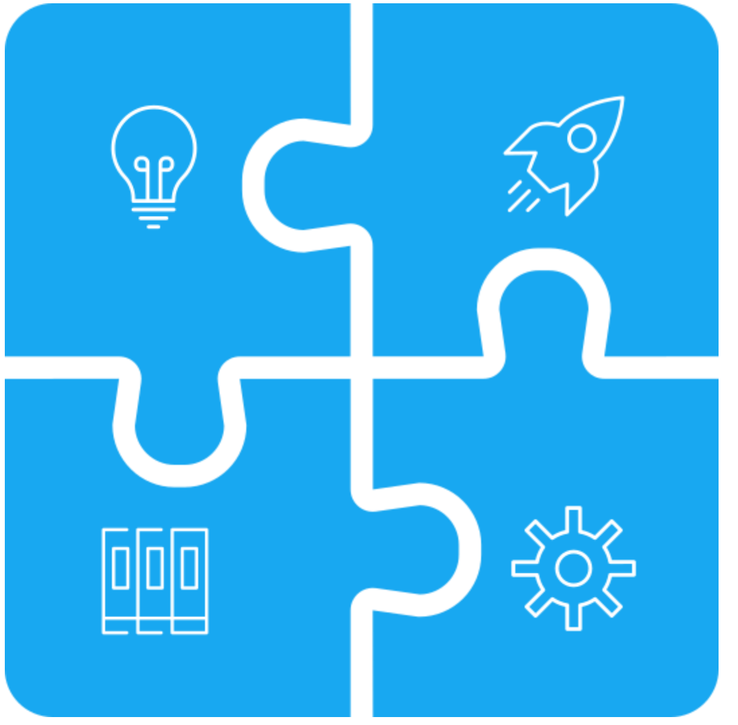
맵에 기본 정보 흡연공간을 설정하기 위하여 공공기관 오픈 api를 사용하였습니다

04

건강한 흡연 앱 설계 목표

흡연자들이 편리하고 건강하게 사용할 수 있는 담배 앱

흡연장소를 지도에 표시된 흡연구역 이외에도 사용자들이 마커를 표시해 추가하고 추가한 마커에 댓글을 달아 서로의 의견을 공유하고 마커한곳까지의 경로를 찾을 수 있도록 도와주는 앱입니다.



05

전체 시스템 구성도

사용자
(user)

시스템을 쓰는 사용자

프론트엔드로 부터 정보를 받고 그것을
조작하여 정보를 주는 사용자

프론트
엔드(UI)

프론트엔드

안드로이드 스튜디오 코틀린을 사용하며
백엔드 와 연동하여 최종적으로 사용자에게
보이고 조작할 수 있는 메인화면을 구현한
시스템

백엔드
(DB)

백엔드

파이어베이스를 사용하여 사용자들의
데이터, 지도상의 데이터 등 데이터를 저장하
여 프론트엔드에게 정보를 넘겨주고 받는
역할을 하는 시스템

06

지난 학기 진행사항

2023-2학기

주차	개발 현황(1-8주차) 및 계획(9-15주차)
1	팀 구성 및 개발 계획
2~3	어플 설치 및 자료조사
4~5	네이버 지도 API연동
6~7	데이터 베이스 연동
8	(중간 발표)
9~10	UI 수정, 공공기관 API 연동
11-12	맵 마커 찍기 기능 추가 , 코멘트 기능 추가
13~14	길 찾기 기능 추가, 맵 마커 삭제 기능 추가
15	1학기 최종발표

07

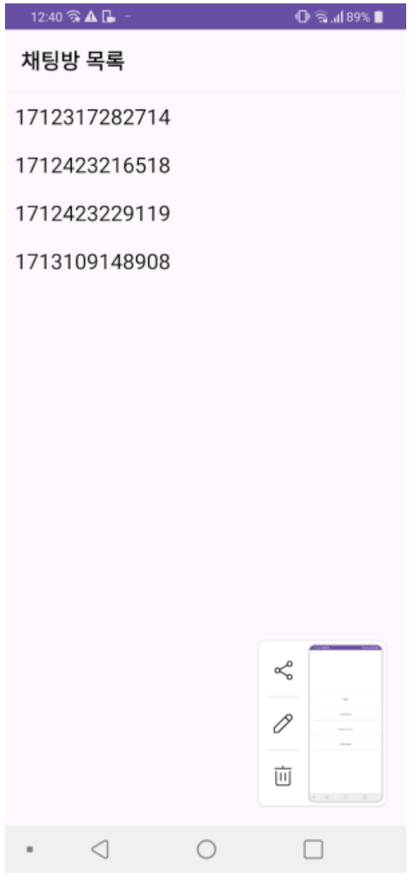
이번 학기 계획

2024-1학기

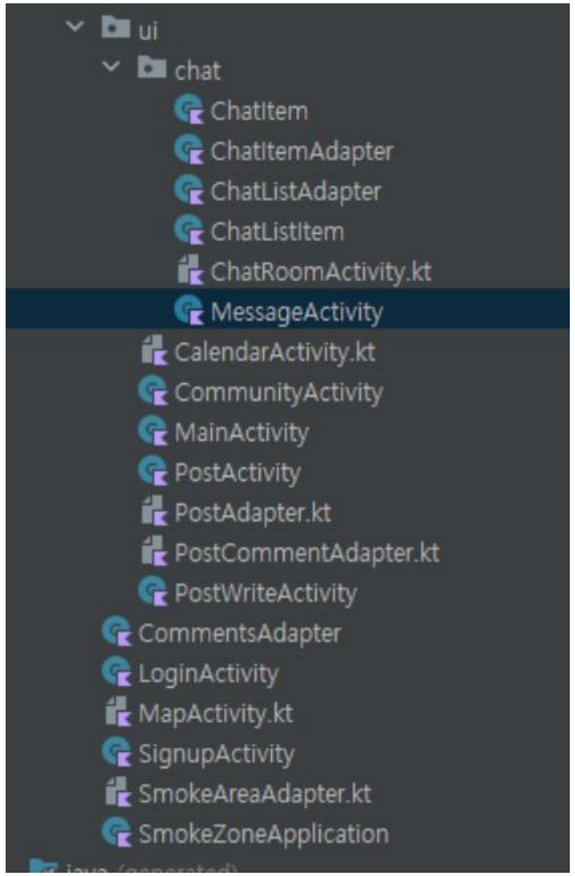
주차	개발 계획
1-2	마커 까지의 길찾기 기능, 현재위치 기반 주변 흡연구역 노출 기능 추가
3-4	커뮤니티 기능, 달력에 언제 흡연했는지 확인하는 기능 추가
5-6	오류 검출 및 코딩 간결하게 수정
7-8	3차발표
9-10	사용자 끼리 일대일 대화 기능, 달력기반 한달, 일년 동안 흡연 총 비용 산출 기능 추가
11-12	사용자들의 금연을 돕기 위한 물품 추천 배너, 핸드폰 팝업바 추가(간단한 흡연체크)
13-14	금연 알람 추가, 한달동안,일년동안 흡연비용과 같은 금액 대 물품 화면 추가
15	최종발표(춘계 작품발표회)

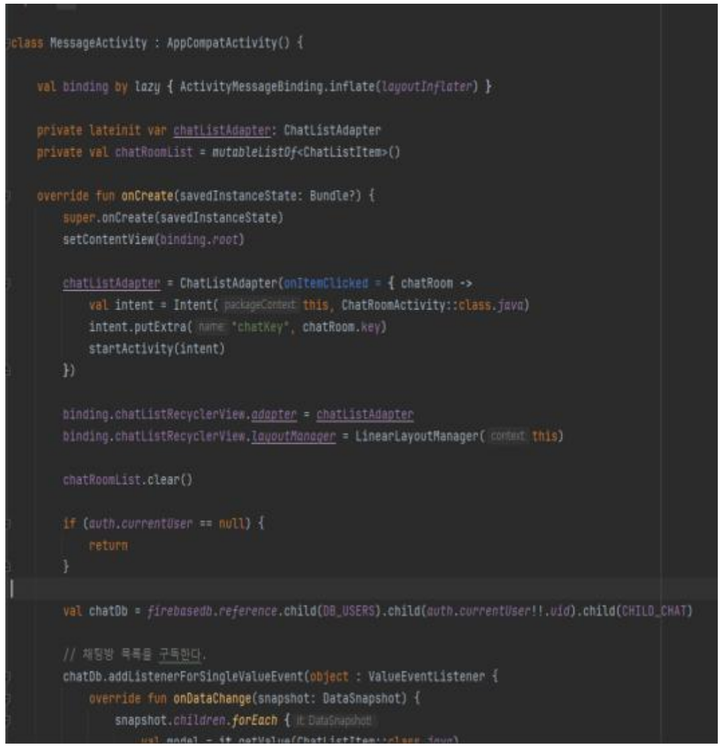
08

쪽지 기능 추가



동작 사진



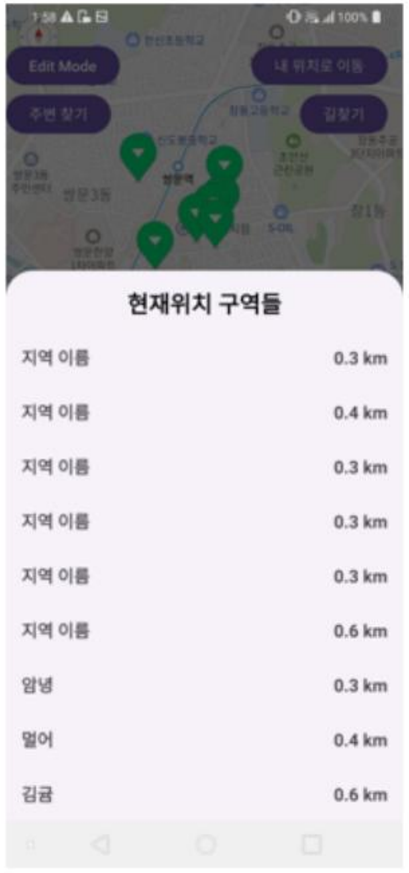


```
class MessageActivity : AppCompatActivity() {

    val binding by lazy { ActivityMessageBinding.inflate(layoutInflater) }

    private lateinit var chatListAdapter: ChatListAdapter
    private val chatRoomList = mutableListOf<ChatListItem>()

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(binding.root)

        chatListAdapter = ChatListAdapter(onItemClicked = { chatRoom ->
            val intent = Intent( packageContext: this, ChatRoomActivity::class.java)
            intent.putExtra( name: "chatKey", chatRoom.key)
            startActivity(intent)
        })

        binding.chatListRecyclerView.adapter = chatListAdapter
        binding.chatListRecyclerView.layoutManager = LinearLayoutManager( context: this)

        chatRoomList.clear()

        if (auth.currentUser == null) {
            return
        }

        val chatDb = firebasedb.reference.child(DB_USERS).child(auth.currentUser!!.uid).child(CHILD_CHAT)

        // 채팅방 목록을 구독한다.
        chatDb.addListenerForSingleValueEvent(object : ValueEventListener {
            override fun onDataChange(snapshot: DataSnapshot) {
                snapshot.children.forEach { it: DataSnapshot
```

코드 소스

09

주변위치 찾기기능 추가



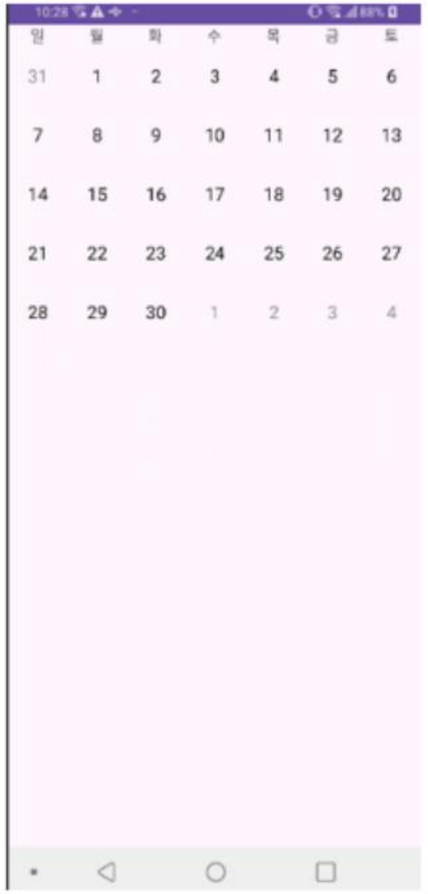
주변 위치까지의 거리 표시

```
        }

        // 현재위치를 가져와서 거리 계산하여 객체 mapping
        fusedLocationClient.lastLocation.addOnSuccessListener { it: Location!
            val currentLocation = it
            adapter.setSmokeAreas(smokeAreaList.map { it: SmokeArea
                it.copy(
                    currentLocation = Location(
                        latitude = currentLocation.latitude,
                        longitude = currentLocation.longitude
                    ),
                    targetLocation = Location(
                        latitude = it.latitude,
                        longitude = it.longitude
                    )
                )
            })

            bottomSheetDialog.show()
        }

    }
}
```

코드 소스

10

달력 기능 추가

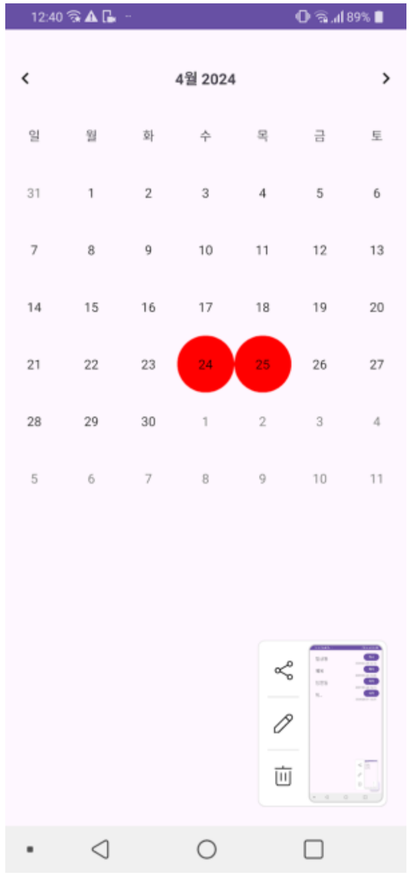


동작 사진

```
import ...

class CalendarActivity : AppCompatActivity() {

    val binding by lazy { ActivityCalendarBinding.inflate(layoutInflater) }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(binding.root)

        binding.calendarView.dayBinder = object : MonthDayBinder<DayViewContainer> {
            override fun create(view: View) = DayViewContainer(view)
            override fun bind(container: DayViewContainer, data: CalendarDay) {
                container.textView.text = data.date.dayOfMonth.toString()
                if (data.position == DayPosition.MonthDate) {
                    container.textView.setTextColor(Color.BLACK)
                } else {
                    container.textView.setTextColor(Color.GRAY)
                }
            }
        }

        val daysOfWeek = daysOfWeek()


        val titlesContainer = findViewById<ViewGroup>(R.id.titlesContainer)
        titlesContainer.children
            .map { it as TextView }
            .forEachIndexed { index, textView ->
                val dayOfWeek = daysOfWeek[index]
                val title = dayOfWeek.getDisplayName(TextStyle.SHORT, Locale.getDefault())
                textView.text = title
            }

        val currentMonth = YearMonth.now()
```

코드 소스

달력에 흡연 기록 기능 추가



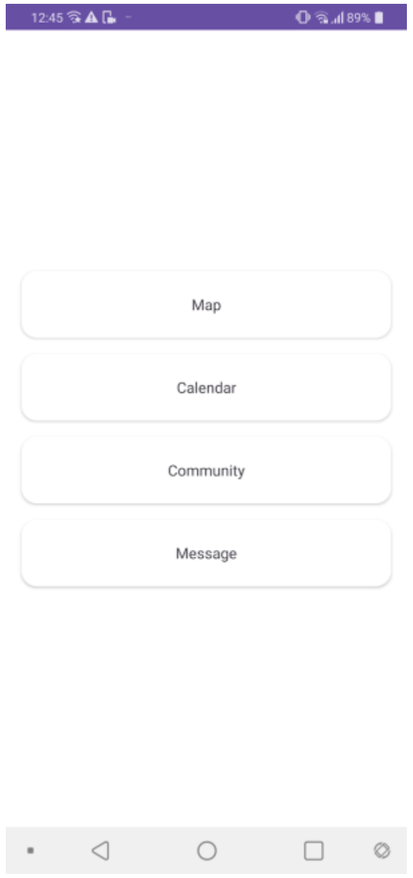
동작 사진

```
private fun bindOnCalendarLongClick() {
    binding.calendarView.setOnDateLongClickListener { widget, date ->
        val newDates = loadDateList( context: this).map { it.toCalendarDay() }.toMutableList()

        val message = if (newDates.contains(date)) {
            "흡연 기록을 제거하시겠습니까?"
        } else {
            "흡연 기록을 추가하시겠습니까?"
        }

        val dialog = AlertDialog.Builder( context: this) AlertDialog.Builder
            .setTitle("기록") AlertDialog.Builder!
            .setMessage(message)
            .setNegativeButton( text: "취소",  listener: null)
            .setPositiveButton( text: "확인") { dialog, which ->
                if (newDates.contains(date)) {
                    newDates.remove(date)
                } else {
                    newDates.add(date)
                }
                saveDateList( context: this, newDates.map { MyDate(it.year, it.month, it.day) })
                widget.invalidateDecorators()
                drawCalendar()
            }.create()

        dialog.show()
        true
    }
}

private fun drawCalendar() {
    val dates = loadDateList( context: this).map { it.toCalendarDay() }

    binding.calendarView.removeDecorators()
```

코드 소스

12

메뉴 목록 기능 추가



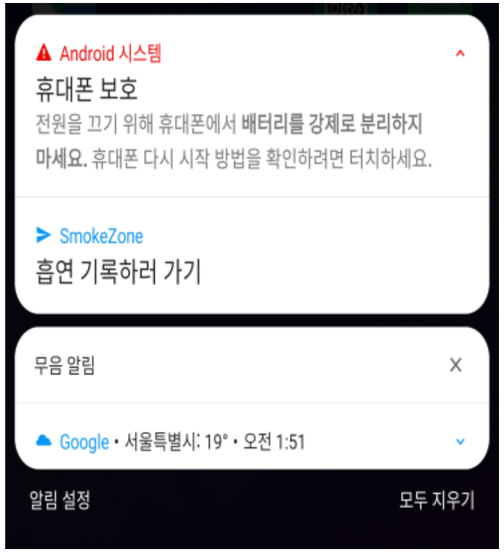
동작 사진

```
class MainActivity : AppCompatActivity() {

    val binding by lazy { ActivityMainBinding.inflate(layoutInflater) }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        setContentView(binding.root)

        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main)) { v, insets ->
            val systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom)
            insets ^setOnApplyWindowInsetsListener
        }

        binding.mapBtn.setOnClickListener { it: View!
            val intent = Intent( packageContext: this, MapActivity::class.java)
            startActivity(intent)
        }

        binding.communityBtn.setOnClickListener { it: View!
            val intent = Intent( packageContext: this, CommunityActivity::class.java)
            startActivity(intent)
        }

        binding.calendarBtn.setOnClickListener { it: View!
            val intent = Intent( packageContext: this, CalendarActivity::class.java)
            startActivity(intent)
        }

        binding.messageBtn.setOnClickListener { it: View!
            val intent = Intent( packageContext: this, MessageActivity::class.java)
            startActivity(intent)
        }
    }
```

코드 소스

13

팝업 화면 ,알람 추가



동작 사진

```kotlin
private val CHANNEL_ID = "CHANNEL_ID"

override fun onReceive(context: Context?, p1: Intent?) {
    context ?: return
    createNotificationChannel(context)

    val notificationManager = ContextCompat.getSystemService(
        context, NotificationManager::class.java) as NotificationManager

    val intent = Intent(context, CalendarActivity::class.java).apply { this: Intent
        flags = Intent.FLAG_ACTIVITY_NEW_TASK or Intent.FLAG_ACTIVITY_CLEAR_TASK
    }

    val pendingIntent = PendingIntent.getActivity(
        context,
        requestCode: 0,
        intent,
        flags: PendingIntent.FLAG_UPDATE_CURRENT or PendingIntent.FLAG_IMMUTABLE
    )

    val notification = NotificationCompat.Builder(context, CHANNEL_ID)
        .setSmallIcon(R.drawable.baseline_send_24)
        .setContentTitle("금연")
        .setContentText("도전하세요!")
        .setPriority(NotificationCompat.PRIORITY_DEFAULT)
        .setContentIntent(pendingIntent)
        .setAutoCancel(false)
        .setShowWhen(false)
        .build()

    notificationManager.notify(Random.nextInt(), notification)
}
```

```kotlin
override suspend fun doWork(): Result {
    // 알림 발생 시 다음날 알림도 발생할 수 있게 설정
    val dailyWorkRequest = OneTimeWorkRequestBuilder<AlarmWorker>()
        .setInitialDelay(getTimeUsingInWorkRequest(), TimeUnit.MILLISECONDS)
        .setId(id)
        .build()

    WorkManager.getInstance(applicationContext).enqueue(dailyWorkRequest)

    val alarmIntent = Intent(applicationContext, AlarmReceiver::class.java).let { intent ->
        PendingIntent.getBroadcast(
            applicationContext,
            requestCode: 0,
            intent,
            PendingIntent.FLAG_IMMUTABLE
        )
    }

    alarmIntent.send()

    return Result.success()
}

private fun getTimeUsingInWorkRequest() : Long {
    val currentDate = Calendar.getInstance()
    val dueDate = Calendar.getInstance()

    dueDate.set(Calendar.HOUR_OF_DAY, LocalDateTime.now().hour)
    dueDate.set(Calendar.MINUTE, LocalDateTime.now().minute)
    dueDate.set(Calendar.SECOND, 0)

    if(dueDate.before(currentDate)) {
        dueDate.add(Calendar.HOUR_OF_DAY, amount: 24)
    }

    return dueDate.timeInMillis - currentDate.timeInMillis
}
```

코드 소스

흡연 비용 비교 물품 화면 추가

동작 사진

```
private fun setSmokePay() {
    val now = LocalDate.now()
    val monthLoadDateList = loadDateList( context: this).filter { it: MyDate
        now.year == it.year && now.monthValue == it.month
    }

    val monthSmokeCount = monthLoadDateList.size
    val monthSmokePay = monthSmokeCount * 5000

    binding.monthSmoke.text = "${formatNumber(monthSmokePay)}원"

    val yearLoadDateList = loadDateList( context: this).filter { it: MyDate
        if (now.year == it.year) {
            true ^filter
        } else {
            now.monthValue <= it.month ^filter
        }
    }

    val yearSmokeCount = yearLoadDateList.size
    val yearSmokePay = yearSmokeCount * 5000

    binding.yearSmoke.text = "${formatNumber(yearSmokePay)}원"

    binding.monthSmoke10Container.isVisible = monthSmokePay >= 100_000
    binding.monthSmoke20Container.isVisible = monthSmokePay >= 200_000

    binding.yearSmoke50Container.isVisible = yearSmokePay >= 500_000
    binding.yearSmoke200Container.isVisible = yearSmokePay >= 2_000_000
    binding.yearSmoke400Container.isVisible = yearSmokePay >= 4_000_000
    binding.yearSmoke1000Container.isVisible = yearSmokePay >= 10_000_000
    binding.yearSmoke2000Container.isVisible = yearSmokePay >= 20_000_000
}
```

코드 소스

구동 시연

KDU Project

THANK YOU FOR ATTENTION

담당자 (9팀)	연락처	최종 발표 일정
최경채, 조성민,김상현	010.2495.5667 (2121058@kdu.ac.kr)	2024.06.11