

Žilinská univerzita v Žiline – Fakulta riadenia a informatiky

Dokumentácia

FakeCall

Martin Uhrina

SZYP21

2022/2023

Letný semester

Úvod

Ako semestrálnu prácu som sa rozhodol vytvoriť mobilnú aplikáciu s názvom FAKE CALL.

Fake Call je mobilná aplikácia, ktorá umožňuje používateľom naplánovať falošný telefonát, ktorý ich zavolá v určenom čase alebo po uplynutí určitého času. Táto aplikácia môže byť užitočná pre rôzne situácie, ako napríklad chcete sa vyhnúť rozhovoru alebo ukázať svojim priateľom, že vám niekto dôležitý volá.

Po preskúmaní Google play som zistil že podobných aplikácií je viacero ale podľa mňa každej niečo chýba, či už sú niektoré neprehľadné, veľmi veľa reklám alebo sú platené.

Preto som sa rozhodol vytvoriť aplikáciu FAKE CALL, ktorá bude čo najviac prehľadná a bude mať viacero dodatočných funkcií, ako napríklad: databázu minulých hovorov, možnosť nahráť si nahrávku, ktorá sa spustí po prijatí hovoru a podobne.

Prehľad aplikácií podobného zamerania

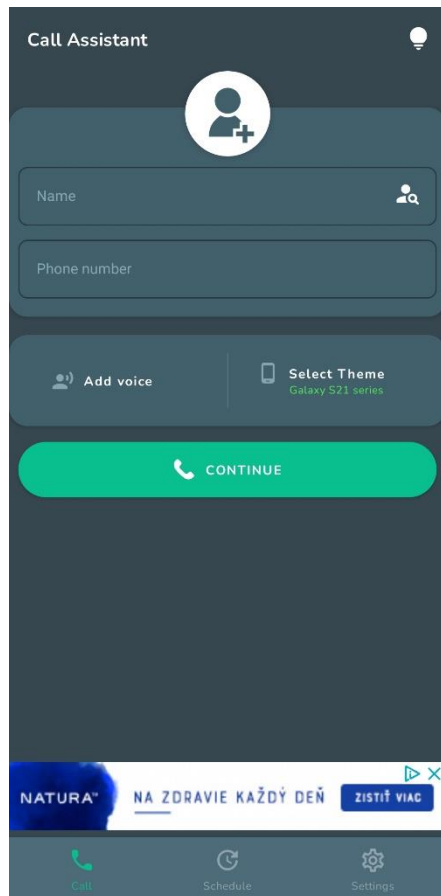
V tejto časti budem opisovať podobné aplikácie, ktoré som našiel na Google play. Opíšem ako fungujú, aké majú funkcie a aký majú dizajn, prípadne čo sa mi tam nepáči. Pre porovnanie som si vybral aplikácie a to Call Assistant a Falošný hovor.

Call Assistant

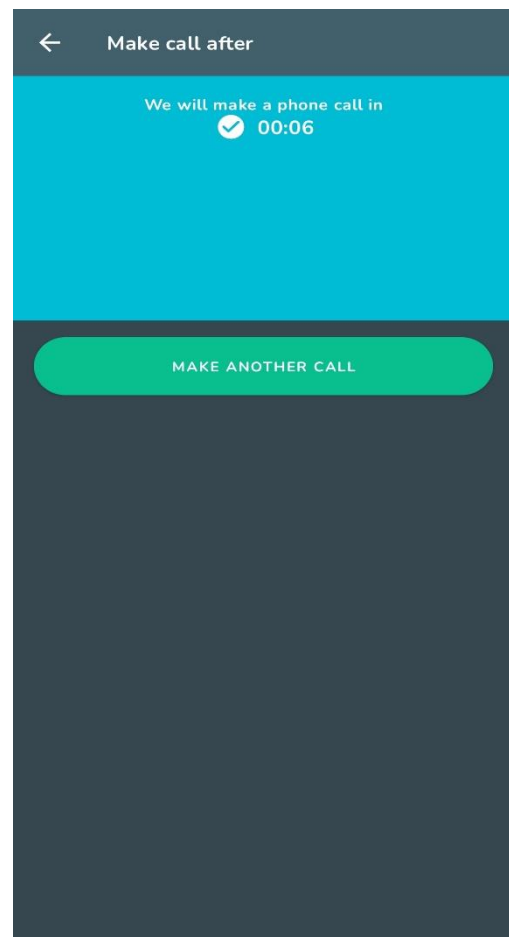
Túto aplikáciu som si vybral pre porovnanie pretože ma celkom oslovila. Na tejto aplikácii sa mi páči dizajn všetkých obrazoviek, ktorý je celkom prehľadný, na obrazovkách nie je príliš veľa icon v ktorých sa človek stráca. V aplikácii sa nachádza veľa doplnkových funkcií ako napríklad: pridanie nahrávky po prijatí hovoru, výber určitého mobilu podľa ktorého sa spustí layout pri zvonení telefónu, pridanie fotografie a podobne. Táto

aplikácia sa mi ako užívateľovi páči ako je navrhnutá a celkovo ako pomerne jednoducho sa používa.

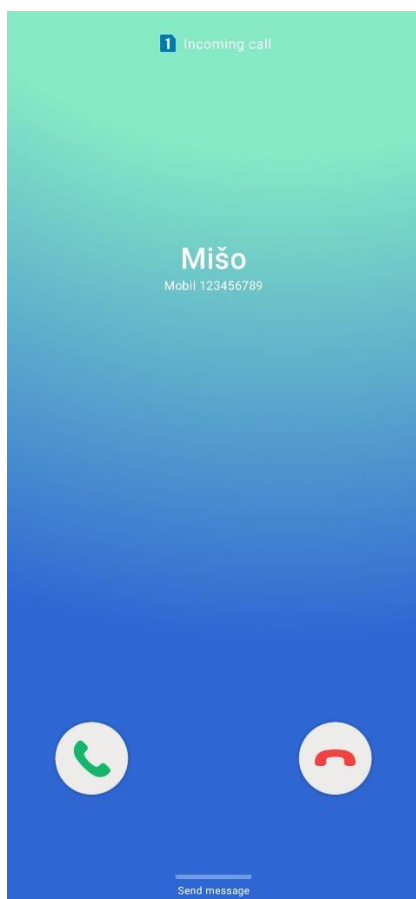
Avšak čo sa mi tejto aplikácií nepáči je veľké množstvo reklám, ktoré dosť kazia príjemný dizajn a pri prechode na ďalšie obrazovky dosť zdržujú.



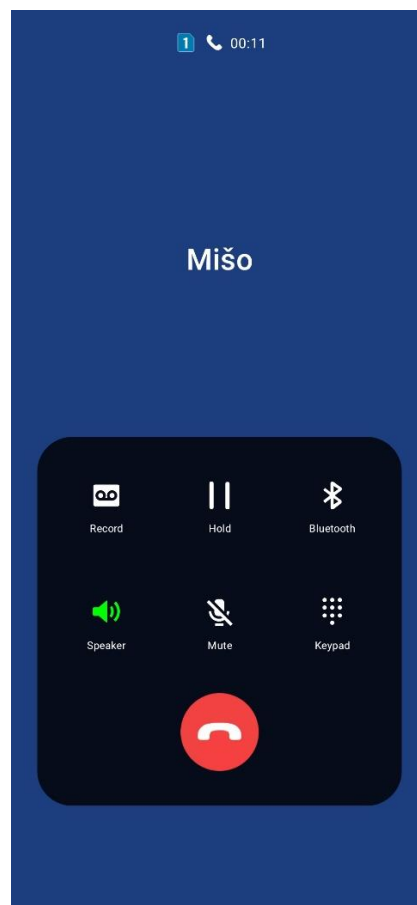
Obrázok 1(Call Assistant)



Obrázok 2(Call Assistant)



Obrázok 3(Call Assistant)

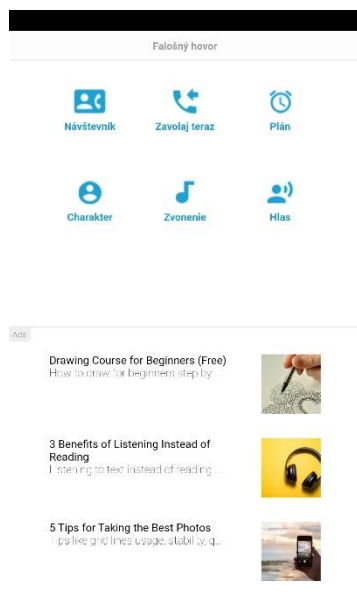


Obrázok 4(Call Assistant)

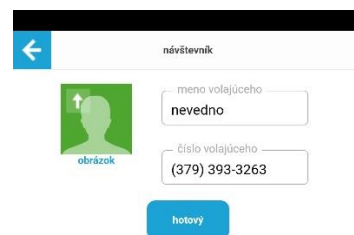
Falošný hovor

Túto aplikáciu som si vybral preto aby som ukázal ako by podľa mňa nemala vyzeráť mobilná aplikácia. Spracovanie tejto aplikácie nie je vôbec user friendly. Aplikácia nemá vôbec pútavý dizajn. Na úvodnom layoute sú icony iba na vrchu a väčšinu obrazovky tvoria reklamy. Aplikácia má možnosti časovača obmedzené a teda doba za koľko príde hovor sú iba 10 sekúnd, 30 sekúnd a 1 minúta.

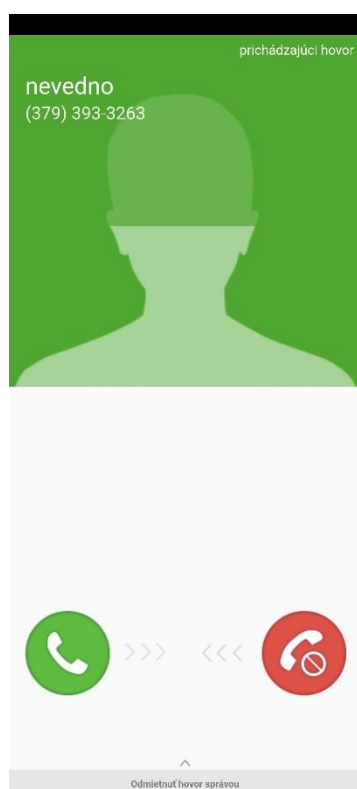
Pozitívum tejto aplikácie je že je zadarmo.



Obrázok 5(Falošný hovor)



Obrázok 6(Falošný hovor)



Obrázok 7(Falošný hovor)



Obrázok 8(Falošný hovor)

Analýza navrhovanej aplikácie

Aplikácia je navrhnutá aby pôsobila čo najjednoduchšie a čo najprehľadnejšie pre užívateľa a tiež aby poskytovala všetky funkcionality, ktoré pomôžu používateľovi.

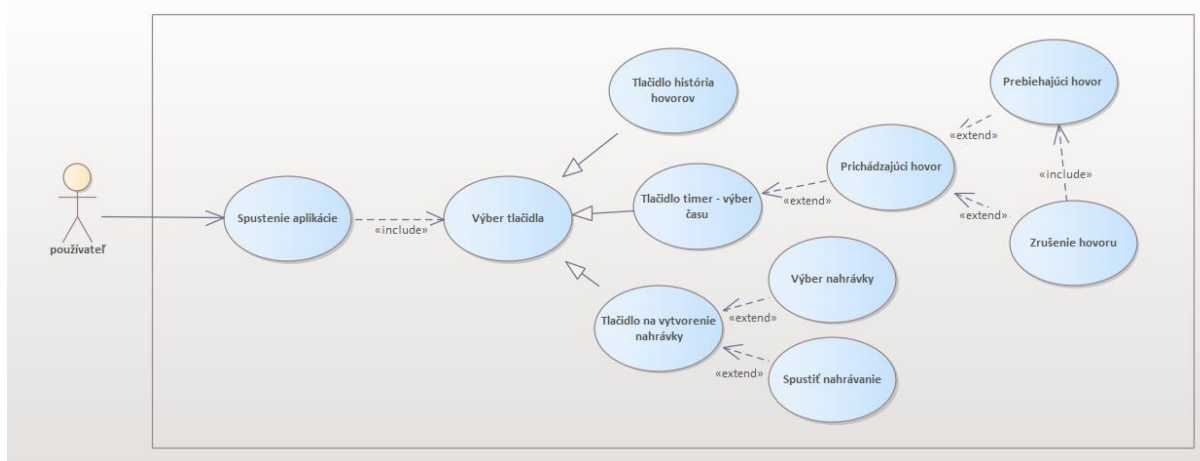
Aplikácia by mala poskytovať tieto funkcionality: vytvoriť prichádzajúci hovor, vytvoriť hovor, spustenie hovorov cez časovač, obsahovať databázu, kde budú uložené predchádzajúce hovory, ktoré sa uskutočnili. Ďalej by aplikácia mohla obsahovať možnosť vytvoriť zvukovú nahrávku, ktorá sa spustí po prijatí hovoru.

Ako je vidieť na návrhoch úvodná obrazovka by mala obsahovať 2 textové polia a 2 buttony. V textových poliach si používateľ napíše od koho chce prijať hovor a z akého čísla. Prvý button bude na posunutie do ďalšieho fragmentu a druhý button bude na skok do activity na vytvorenie nahrávky.

V menu môžu byť 2 položky a to "história" a "O aplikácii".

Na ďalšej obrazovke si používateľ zvolí za aký čas chce prijať hovor. Možnosti by mali byť navrhnuté cez radio-buttons. Následne po stlačení buttonu v dolnej časti obrazovky sa spustí časovač a po vypršaní časovača sa spustí hovor.

Posledné 2 obrazovky by mali vyzeráť ako prichádzajúci hovor s 2 tlačidlami, prijať hovor a odmietnuť hovor. Po prijatí hovoru sa zavolá aktivita kde bude prebiehať hovor.



Popis implementácie

1. Implementácia časovača po prijatí hovoru, tento časovač zobrazuje ako dlho trvá hovor.

```
private fun startTimer()
{
    var seconds : TextView = findViewById(R.id.seconds)
    var minutes : TextView = findViewById(R.id.minute)
    var countSec: Int = 0
    var countMin: Int = 0

    val cTimer = object : CountdownTimer( millisInFuture: 2000000, [countDownInterval: 1000] ) {
        override fun onTick(millisUntilFinished: Long) {
            countSec++
            if (countSec == 60)
            {
                ...
            }
            if (countSec<10)seconds.text = "0" + countSec.toString()
            else seconds.text = countSec.toString()

        }

        override fun onFinish() {

        }

    }

    cTimer.start()
}
```

2. Implementácia holdera pre zobrazovanie dát z Room databázy.

```
class CallViewHolder(view : View) : RecyclerView.ViewHolder(view) {
    private var callName: TextView? = null
    private var callNumber: TextView? = null
    private var callImage : ImageView? = null
    private var callData : TextView? = null

    init {
        callName = view.findViewById(R.id.callName)
        callNumber = view.findViewById(R.id.callNumber)
        callImage = view.findViewById(R.id.imageViewInRecycle)
        callData = view.findViewById(R.id.callDate)
    }

    fun setCallData(calldata: CallData) {
        callName?.text = calldata.meno
        callNumber?.text = calldata.cislo
        callImage?.setImageResource(R.drawable.acceptcall)
        callData?.text = calldata.currentDate
    }
}
```

3. Využite live data vo fragmente TimerFragment

```
class TimerFragment : Fragment() {
    // TODO: Rename and change types of parameters
    private var param1: String? = null
    private var param2: String? = null
    var counterRemainingTime = MutableLiveData<Long>()
    var unitToEnd = MutableLiveData<Long>()
    var nameLive = MutableLiveData<String>()
    var numberLive = MutableLiveData<String>()
    var isOnStop = MutableLiveData<Boolean>()
    var countQuantity = MutableLiveData<Int>()

    polconero
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        arguments?.let { it: Bundle |
            param1 = it.getString(ARG_PARAM1)
            param2 = it.getString(ARG_PARAM2)
            counterRemainingTime.value = 0
            unitToEnd.value = 10
            isOnStop.value = true
            countQuantity.value = 0
        }
    }

    val REQUEST_PHONE_CALL = 1
    var POMOC = false
    polconero
    override fun onCreateView(
```

4. Vymazávanie nahrávok

```
private fun deleteThis() {
    var deletedFile : String = String()
    val builder = AlertDialog.Builder(context: this)
    val inflater : LayoutInflater = LayoutInflater
    val dialogLayout = inflater.inflate(R.layout.edit_text_layout, root: null)
    val editText = dialogLayout.findViewById<EditText>(R.id.et_editText)
    with(builder)
    { this: AlertDialog.Builder
        setTitle("Insert deleted file")
        setPositiveButton(text: "DELETE") { dialog, which ->
            deletedFile = editText.text.toString()
            var file = File(pathname: "/storage/emulated/0/Android/data/com.example.fake_call/files/storage/emulated/0/"+ deletedFile + ".mp3")
            if(!file.exists())
            {
                Toast.makeText(context, text: "This file not exist", Toast.LENGTH_SHORT).show()
            }
            else
            {
                Toast.makeText(applicationContext, text: "file was deleted", Toast.LENGTH_SHORT).show()
                file.delete()
                refreshListView()
            }
        }
        setNegativeButton(text: "Cancel") { dialog, which ->
            Log.d(tag: "Main", msg: "Negative")
        }
        setView(dialogLayout)
        show() ^with
    }
}
```

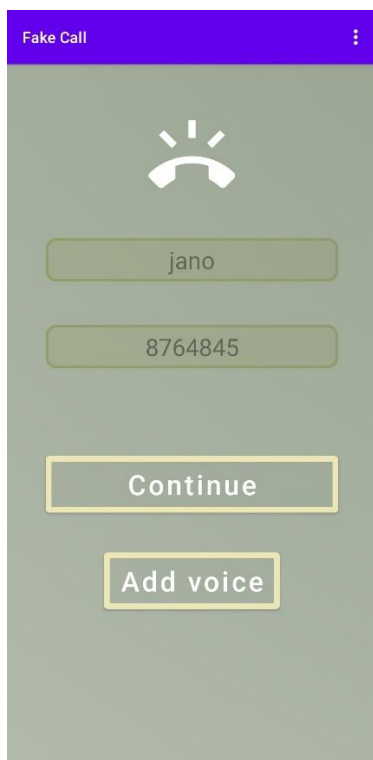

5. Vzýžitie Data Bindingu pri vyberaní časovača

```
//timer
if (!POMOC)
{
    counterRemainingTime.value = 0
    val function = binding.idFunction.checkedRadioButtonId
    var timerID = binding.idChipGroup.checkedChipId
    var timer: Long = 10
    when(timerID){
        binding.idteraz.id -> timer = 10
        binding.id10.id -> timer = 10000
        binding.id20.id -> timer = 20000
    }
    when(timerID)
    {
        binding.idteraz.id -> counterRemainingTime.value = 10
        binding.id10.id -> counterRemainingTime.value = 10000
        binding.id20.id -> counterRemainingTime.value = 20000
    }

    POMOC = true
    // startTimer(longTimer = timer, binding, telNumber, name)
    startTimer(longTimer = counterRemainingTime.value!!, binding, telNumber, name)
}

//star call
ActivityCompat.requestPermissions(
    requireActivity(),
    arrayOf(Manifest.permission.CALL_PHONE, Manifest.permission.RECORD_AUDIO),
    REQUEST_PERMISSIONS_CALL_PHONE,
    null)
```

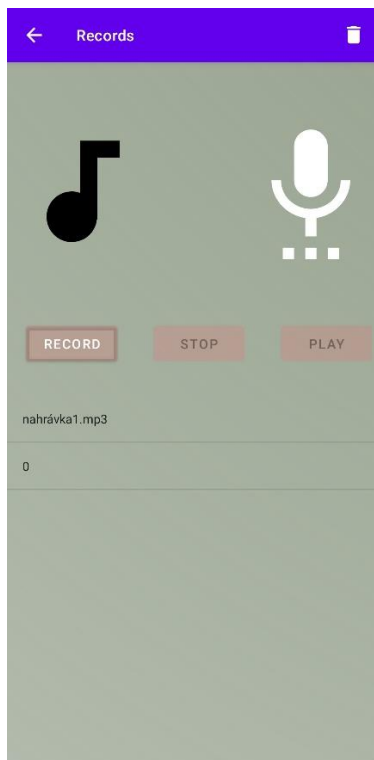
Dizajn aplikácie



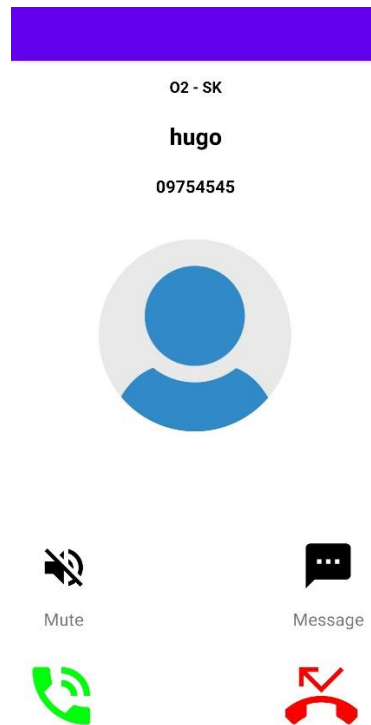
(First Layout)



(Timer layout)



(Voice layout)



(Incoming Call)

Zdroje

<https://developer.android.com/docs>

<https://kaveerrajcoomar.blogspot.com/2017/05/fake-call-androidapplication.html>

<https://skillmea.sk/blog/learn2code-konci>

<https://stackoverflow.com/>