

# REVIVER

Muntoiu Adrian-Andrei & Crisan Mihai-Tudor

Prof. Coordonator Camenița Daniela & Gal Carmen



# ○ CUPRINS

1. CE ESTE **REVIVER** ?
2. DE CE AM CREAT ACEASTĂ APLICAȚIE ?
3. DE CE AȚI ALEGE ACEASTĂ APLICAȚIE ?
4. CUM FUNCȚIONEAZĂ APLICAȚIA REVIVER ?
5. ALTE ELEMENTE ALE APLICAȚIEI
6. BIBLIOGRAFIE

## CE ESTE REVIVER ?

Reviver este o aplicație Android concepută pentru a ajuta utilizatorii să limiteze și să controleze utilizarea aplicațiilor prin intermediul unor restricții inteligente bazate pe suprapuneri. Indiferent dacă încercați să reduceți timpul petrecut în fața ecranului, să reduceți distragerile sau să mențineți obiceiuri digitale sănătoase, Reviver oferă moduri de control flexibile și aplicabile pentru a vă menține concentrați și a vă folosi timpul cu atenție.

## DE CE AM CREAT ACEASTA APLICATIE ?

Pe măsură ce telefoanele au devenit din ce în ce mai esențiale pentru viața cotidiană, utilizarea excesivă a aplicațiilor, în special rețelele de socializare și jocurile de orice fel, a devenit o problemă gravă, mai ales pentru tineret.

Deși există funcții native de bunăstare digitală, acestea sunt ușor de ocolit sau au o funcționalitate limitată, și, cu examenul de bacalaureat după colț, timpul liber devine din ce în ce mai redus, iar reviver se asigură ca profiți de acesta, fără a elimina complet telefonul din viața utilizatorului.

Pe lângă că am vrut să creăm o aplicație care să rezolve aceste probleme, mereu ne-am dorit să lucrăm la o aplicație Android, iar acest proiect a fost soluția perfectă pentru a îndeplini ambele dorințe.

## DE CE ATI ALEGE ACEASTA APLICATIE ?

Reviver oferă o flexibilitate unică în controlul timpului petrecut pe telefon, fiind mai mult decât o simplă aplicație de limitare. Spre deosebire de alte aplicații care impun restricții rigide, Reviver îți permite să alegi modul de control potrivit stilului tău: poți seta un timp limită, un număr maxim de deschideri, o protecție cu parolă sau o suprapunere constantă. În plus, aplicația este complet personalizabilă pentru fiecare aplicație în parte, oferă o interfață intuitivă, este tradusă în mai multe limbi și continuă să funcționeze discret în fundal, chiar și după ce a fost închisă. Cu Reviver, obții echilibrul perfect între libertate și disciplină digitală, într-un mod eficient.

# ○ CUM FUNCTIONEAZA APLICATIA REVIVER

Programul este scris în limbajul de programare **Kotlin**, folosește o interfață făcută în **Figma**, care a fost construită apoi cu fișiere de format XML, apoi a fost asamblată în **Android Studio**. Aplicația Reviver este formată din 5 fișiere importante. Aceste fișiere importante sunt **MainActivity.kt**, **MonitoringService.kt**, **Overlay.kt**, **AppDetails.kt** și fișierul de configurare **ReviverPrefs.xml**.

Inițial, funcția **onCreate** este apelată și verifică dacă aplicația are permisiunile necesare, apoi apelează funcția **startMonitoringService** care accesează fișierul **MonitoringService.kt**. Permisiunile necesare care fac această aplicație să funcționeze sunt permisiunea **"ACTION\_USAGE\_ACCESS\_SETTINGS"** care permite aplicației Reviver să acceseze aplicațiile telefonului prin intermediul setărilor, și permisiunea **"ACTION\_MANAGE\_OVERLAY\_PERMISSION"** care ne lasă să adăugăm o suprapunere peste aplicațiile telefonului. În funcția **onCreate** reîncărcăm aplicațiile adăugate din fișierul *ReviverPrefs.xml* și inițializăm elementele grafice, precum butonul de adăugare a unei aplicații și meniul de navigare.

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

    /// Trebuie sa avem permisiunile necesare ca sa deschidem aplicatia
    checkAndRequestOverlayPermission()
    if (!hasUsageStatsPermission()) {
        requestUsageStatsPermission()
    } else {
        startMonitoringService()
    }

    loadAndDisplaySelectedApps()
    /// Butonul de adaugare a unei aplicatii acceseaza functia
    /// showAppSelectionDialog
    val addButton: Button = findViewById(R.id.addButton)
    addButton.setOnClickListener {
        showAppSelectionDialog()
    }

    /// Aici schimbam elementele unei pagini din meniu
    val bottomNavigationView = findViewById<BottomNavigationView>
(R.id.bottomNavigationView)
    bottomNavigationView.setOnItemSelectedListener { menuItem ->
        when (menuItem.itemId) {
            R.id.nav_home -> {
                replaceFragment(HomeFragment())
            }
            R.id.nav_stats -> {
                replaceFragment(StatsFragment())
            }
            R.id.nav_info -> {
                replaceFragment(InfoFragment())
            }
            R.id.nav_settings -> {
                replaceFragment(SettingsFragment())
            }
        }
    }
    true
}
```

Butonul de adăugare a unei aplicații este legat de funcția **showAppSelectionDialog**, unde un meniu de adăugare modificat ne lasă să adăugăm ce aplicații dorim să adăugăm folosind casete de selectare. După ce am selectat o aplicație din meniu, funcția **addAppToMainLayout** adaugă aplicația în fișierul **ReviverPrefs.xml**.



Odată ce am adăugat o aplicație, dacă apăsăm aplicația în sine, suntem trimiși la funcția **editAppDetailsDialog**, care ne lasă să modificăm trăsăturile aplicației adăugate. Trăsăturile aplicației adăugate sunt salvate în fișierul **ReviverPrefs**. Când edităm aplicația, avem 4 casete de dialog, precum limita de timp, numărul maxim de deschideri, modul de afișare a suprapunerii și setarea parolei. Cele 3 moduri de afișare a suprapunerii sunt:

- **Modul 1**, care afișează o suprapunere după numărul de secunde din caseta de limită de timp;
- **Modul 2**, care afișează o suprapunere după ce deschidem aplicația de mai multe ori decât numărul de deschideri din caseta de număr maxim de deschideri;
- **Modul 3**, care afișează o suprapunere cu o parolă impusă de utilizator după numărul de secunde din caseta de limită de timp,
- **Modul 4**, care afișează constant o suprapunere după ce limita de timp a fost depășită

Fișierul **MonitoringService.kt** începe funcția **onCreate** și funcția **monitorTask**. În funcția **monitorTask**, sunt apelate funcțiile **checkAndResetDailyLaunchCounts** pentru **modul 2**, care resetează numărul de deschideri după 24 de ore, și funcția **monitorAppUsage**.

caz că aplicația adăugată apare în **ReviverPrefs**, verificăm în ce mod apare, și funcția apelează funcțiile **checkAndUpdateAppLaunches**, **handleTimeLimit**, **handlePasswordMode** sau **handleConstantOverlay** în funcție de diferitele moduri.

Fișierul **Overlay.kt** se ocupă cu afișarea suprapunerii. În acest fișier există câteva funcții importante, precum **showOverlay**, **setupRandomCloseButton** și **removeOverlay**. În funcția **showOverlay** se inițializează butoanele. Pe lângă aceste inițializări, funcția verifică în ce mod este aplicația adăugată, ca să poată schimba vizibilitatea elementelor, și accesează funcția **setupRandomCloseButton**.

Funcția **setupRandomCloseButton** inițializează butonul de închidere a suprapunerii într-un colț diferit, ca să nu lase utilizatorul să închidă suprapunerea din același loc de fiecare dată.

```
fun showOverlay(private val context: Context){
    if (overlayView != null) return

    when (app?.mode) {
        "Mode 3 (Password Protected)" -> { /// Modul 3
            submitButton.setOnClickListener {
                val input = passwordInput.text.toString()

                if (input == app.password) {
                    removeOverlay()
                } else {
                    Toast.makeText("Wrong password!")
                        passwordInput.text.clear()
                }
            }
            forgotPassword.setOnClickListener {
                removeOverlay()
                sendToAppDetails(app)
            }
        }

        "Mode 2 (Launch Limit)" -> { /// Modul 2
            messageTextView.text = "$message\nCurrent opens:
            "${app.currentOpens}/${app.maxOpens}"
            setupRandomCloseButton(closeButton)
        }

        else setupRandomCloseButton(closeButton)
        /// Modul 1 si Modul 4
    }
}
```

```
when (getModeType(app.mode)) {
    1 -> handleTimeLimit(app)
    2 -> checkAndUpdateAppLaunches(app)
    3 -> handlePasswordMode(app)
    4 -> handleConstantOverlay(app)
    else -> handleTimeLimit(app)
}
```

## ○ ALTE ELEMENTE ALE APLICATIEI

Reviver nu se rezumă doar la suprapuneri: studiile arată că între 25 % și 38 % din populație manifestă deja utilizare problematică a telefonului mobil, ceea ce subliniază necesitatea unei aplicații ca Reviver, iar în Statele Unite utilizatorii petrec în medie aproape cinci ore pe zi pe smartphone.

Pentru a sprijini utilizatorii din România, Franța, Germania, Spania și SUA, am adăugat opțiunea de schimbare a limbii în aplicație, astfel încât fiecare să poată naviga în propria limbă și să înțeleagă cu ușurință setările. Mai mult, funcția „tipul zilei” oferă idei simple și inspiratoare despre ce ai putea face astăzi în loc să-ți folosești telefonul - de la plimbări scurte la exerciții de relaxare - pentru a transforma timpul economisit într-o oportunitate reală de reconectare cu viața offline.

## ○ BIBLIOGRAFIE

Documentare Kotlin:

<https://www.w3schools.com/KOTLIN/index.php>

<https://kotlinlang.org/docs/kotlin-pdf.html>

<https://adapty.io/blog/kotlin-vs-java>

Documentare Android:

<https://www.geeksforgeeks.org/android-studio-tutorial/>

<https://developer.android.com/codelabs/basic-android-kotlin-compose-first-app>

<https://www.geeksforgeeks.org/application-manifest-file-in-android/>

<https://www.youtube.com/watch?v=FjrKMcKahY&t=386s>

<https://www.youtube.com/watch?v=jOFLmKMOcK0>

[https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV\\_yJ-E](https://www.youtube.com/playlist?list=PLS1QulWo1RIbb1cYyzZpLFCKvdYV_yJ-E)

<https://www.youtube.com/watch?v=jOFLmKMOcK0>

<https://www.youtube.com/watch?v=JOdWT50bWw4>

<https://www.youtube.com/watch?v=plnLs6aST1M>

<https://www.figma.com>

<https://www.builder.io/blog/figma-to-svelte>

Documentare XML și resurse

<https://chatgpt.com/>

<https://deepseek.com/>

[https://www.w3schools.com/xml/xml\\_what\\_is.asp](https://www.w3schools.com/xml/xml_what_is.asp)