

# Daily Routines fejlesztői dokumentáció

Dokumentáció a 2016-2017 II. Alkalmazások fejlesztése kurzushoz beadandó projekthez. Készítette: Keresztes Dániel

2017.05.15

## A program általános leírása

A Daily Routines személyes igényekben gyökerező ötletként indult. Szükségem volt egy programra, ami segít kialakítani rutinokat úgy, hogy közben arra is figyel, nehogy elkésssek. Első nekifutásra a Pebble okosórán fejlesztettem egy prototípust, amit jó ideig használtam is, de hamar kevés lett a platform az általam kivitelezni szándékozott célokhoz – arról nem is beszélve, hogy a cég megvétele körüli bizonytalanságok eléggnek tűntek ahhoz, hogy ne erőltessem a platformot.

Az Android projektnek január közepén álltam neki, és a következő pár hónapban az iskola mellett dolgoztam rajta. Jelenleg nincs még publikálhatónak szánt állapotában, további munkát igényel. Mindenesetre kisebb hibáktól eltekintve használható állapotú.

## Fejlesztői környezet és dependenciák

A projekt kezdetén úgy döntöttem, hogy az elsődleges prioritás egy használható prototípus mielőbbi elkészítése. Emiatt a régebbi, 5.0-ás Androidnál régebbieket futtató telefonokat nem támogatja pillanatnyilag az app. A praktikus megfontolás emögött az volt, hogy a projekt komplexitását nem akartam *support* könyvtárakkal idejekorán növelni. Ugyanakkor azt gondolom, hogy, ha szükséges, akkor ezen funkcionalitás a későbbiekben hozzáadható.

A programírást és tesztelést segítő, a legfrissebb stabil Android Studio-t – pillanatnyilag 2.3.2 – használom fejlesztői környezet gyanánt.

Külső dependenciákat az app pillanatnyilag nem használ, kivéve a **Firebase**-t, amiről majd alább ejtek több szót.

A program verzióját szemantikus verziózással adom meg (nagy\_verzió.kis\_verzió.patch) a felhasználó számára, belsőleg pedig egész számokkal követem. Az app jelenleg a 0.5.7-es verzió van.

A verziókövető rendszer, amit használok, Bitbucketen hosztolt git, ahol a master ágon van mindig a stabil verzió, a többi ágon pedig *feature*-öket valósítok meg, amiket később visszamergelek masterre.

A teszt környezetem egy Android 6.0-t futtató második generációs Motorola Moto X. Ezen kívül számos más eszközön fut, de statisztikailag nem jelentős méretű ezek száma.

### **A projekt managementje, tervezés**

A projekt sínen tartása érdekében a tervezési fázisban folyamatábrákat és UML diagrammokat készítettem (a *docs* mappában megtalálhatóak). Később kellőképpen elrugaskodtam ezektől, nem feltétlenül tükrözik az aktuális állapotokat.

A projekt managementjéhez a Trello-t használom, kanban módszerrel rendszerezem a feladatokat, ahol a különböző területeknek megvan a színkódja – például a BUGFIX-eket pirossal jelölöm, az adminisztrációs feladatokat zölddel, stb.

### **A program felépítése, látható funkciói**

A program futása során a felhasználó a kezelő felülettel érintkezik. A következő funkciók betöltésére használ pillanatnyilag különböző képernyőket az alkalmazás:

- Rutinok képernyő, ahol ki lehet választani már meglévő rutint, el lehet jutni a beállításokba, valamint hozzá lehet adni új rutint
- Beállítások
- Rutin profil, ahol hasznos információkhoz lehet hozzájutni a használt rutinok futási statisztikáihoz, el lehet indítani, de szerkeszteni is a rutint
- Hozzáadás/szerkesztés
- Rutin óra, ami a program fő képessége

A fentiek működése többnyire magától értetődő, kivéve talán az órát. Az óra célja, hogy megakadályozza, hogy elcsússzunk egy rutinnal, miközben tájékoztat arról, hogy mi kellene éppen csinálnunk. A nagy számláló az éppen aktuális elem idejét mutatja, az alatta lévő kisebb pedig az eddig megspórolt időt.

Ha az elem ideje lejár, akkor elkezd a maradék időt felhasználni. Ha ebben a számlálóban sem marad már semmi, akkor negatívba kezd el futni. Amikor végre tovább tudunk lépni, a program automatikusan elosztja a hátralévő elemek között arányosan az így elvesztett időt.

Ha minden időnk elfogy, az óra jelzi, hogy vége a rutinnak. Ezek után frissíti a profil képernyőn a statisztikákat, hogy később optimalizálni tudjuk rutinunkat.

### **A program technikai kivitelezése**

#### **Adatok tárolása**

A program működéséhez számos adatot kell perzisztensen tárolni. Ezek egy része gyorsan elérhető kell, hogy legyen, de nem jelentős méretű, míg a többi méretre nagyobb, de nem kell annyiszor elérni.

- Beállítások tárolása az Android rendszer által biztosított kulcs-érték páron alapuló gyors háttértárban valósul meg. Itt pillanatnyilag a rezgés, az notifikációk és a záróképernyő előtt látszódó rutin-óra preferenciáit tárolom.
- A felhasználó által bevitt rutinok adatai jelentősebb mennyiségű adatot jelentek, mint a beállítások. Ennek megfelelően az Android operációs rendszer által nyújtott SQLite adatbázist használok külön ORM réteg nélkül. Az adatbázist egy *ContentProvider* implementáció segítségével érem el, hogy ne kelljen a felhasználói felülettel foglalkozó kódból nyers SQL-t meghívnom. A különböző adatbázis bejegyzéseket és funkcionalitásokat egy *szerződésben* definiált URI-k alapján azonosítom.

### Az adatbázisban tárolt adatok elérése

Nem szerencsés a fő alkalmazás szálon elérni az adatbázist, mivel viszonylag hosszú időbe is telhet, mire az használható választ ad. Emiatt minden adatbázis olvasás párhuzamos szálon fut mindegyik, azt igénybe vevő képernyőn.

### Időzítések

A program egyik jelentősebb képessége, hogy be lehet állítani egy időpontot, amire szeretnék befejezni a rutint. Ekkor az alkalmazás kiszámítja a kezdés ideális időpontját, és ha nincs kikapcsolva a beállításokban, akkor egy értesítéssel jelez a fenti tényről. Ennek megoldása valamivel bonyolultabb, mint amire elsőként következtethetnénk.

Egyrészt, és elsősorban, ha a rendszerre bízunk ezen események időzítését – és igen tanácsos így tenni, mivel vagy túlságosan nagy erőforrásokat emészteneünk fel folyamatos *wake-lockkal*, vagy különé a rendszer az alkalmazásunkat – akkor valamilyen szintű bizonytalansággal számolnunk kell. Az Android rendszer az energiatakarékosság jegyében hajlandó késleltetni, vagy korábban felhasználni a fejlesztő által kért ébresztést.

Másrészt két jelentős esemény van, ami keresztül húzhatja számításainkat, amikor előre beállított értesítésekről van szó: újraindulás, és az alkalmazás leállítása (beállításokból le lehet úgy állítani egy appot, hogy az nem futhat, amíg kézzel vagy másik appal el nem indítjuk). Ezeket kiküszöbölve, elvileg mindig meg kell, hogy kapjuk az értesítéseinket.

### Óra háttér működése

Az óra folyamatos működtetése számos technikai kihívást jelentett, mivel egyes esetekben a képernyő kikapcsolásával minden magasabb szintű program futása megáll – feltéve, hogy egy másik alkalmazás nem birtokol *wake-lock*-ot. Ekkor az óra visszaszámlálója megáll, sem számolni, sem rezegni nem tud.

Pillanatnyilag ezt a problémát nem túl elegánsan, *wakelock*-kal oldom meg. Természetesen ez a közeljövőben kijavításra kerül a többi nagyobb bug megoldásával egyetemben.

## Jövőbeli tervek

- A fenti ébrenléti problémát prioritásom kijavítani.
- Az alább ismertetett pontatlansági bugot megoldom pontosabb időábrázolással
- Statisztikai funkciók
- “Segítség a használatához” képernyő implementálása
- Különböző marketing anyagok összeállítása
- Github-page weblap összeállítása
- Google Play követelmények kielégítése és publikáció

## Támogatás, bugfixek

Pillanatnyilag a **Firebase** hibakövető szolgáltatását használom a bugok távoli figyelésére. Ez az egyetlen aspektusa a programnak, ami a hálózati eszközöket használhatja. A publikálás után egy ideig mindenképpen tervezek hibajavításokat vállalni, amíg ésszerű.

A megjelenés után Githubra fog átkerülni a projekt GPL licenccel, úgyhogy az ottani hibakövetőt fogom majd használni, ha lesz rá felhasználói igény.

## Ismert problémák

- Kisebb UI következetlenségek
- Pontatlanság: vélhetően az Android rendszer időzítés-kezelésének, valamint az implementáció hibáiból fakadóan a program általában pár másodperccel előbb befejezi a rutin futását (több elem esetében, hosszabb távon észrevehetőbb), mint a saját kiírása, és idő szerint kellene. Bár túlmenni a beállított kereten nem enged, kellemetlen lehet, ha marad még felhasználható ideje a felhasználónak, de a program nem fut végig.