Documentatia tehnica | Focus: Time Keeper

Cerinte sistem minime:  
• Android 5.1+ (API 22+) (Lollipop+)  
• Conexiune la internet. (Viteza: 5KB +)  
• Memorie ram: 250 MB+  
• Memorie stocare: 15 MB+  
• CPU: ARM Cortex-A53, 1300 MHz sau Snapdragon 630 sau echivalent

Cerinte sistem optime:  
• Android 6.0+ (API 23+) (Marshmallow+)  
• Conexiune la internet: 50KB+  
• Memorie ram: 1GB+  
• Memorie stocare: 30MB+  
• CPU: Snapdragon 730 sau echivalent

Arhitectura:

* 24 fisiere ce contin cod Java (back-end) dintre care: 3 reprezinta clase prin care se transmit date
* 37 fisiere ce contin cod XML (UI)

Fisiere Java (majoritar contin o singura clasa, cu acelasi nume) listate in ordine alfabetica, apoi listate functiile din fiecare clasa:

* ActivityDataFieldsActivity:
* Sumar clasa: Prin intermediul acestei clase si functiile asociate acesteia utilizatorul introduce o activitate intr-un program
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* checkDigit(@Number): rezolva probleme de afisare ale numerelor (pentru ore si minute)
* changeSelectedImage(): functie nefolosita inca, va fi folosita in urmatoarele iteratii
* BottomNotificationBar:
* Sumar clasa: Arata o notificare in josul oricarei activitati
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* Class\_FileLocations:
* Sumar clasa: Folosita pentru locatiile fisierelor aplicatiei
* Nu contine functii
* Class\_PassActivityDetails:
* Sumar clasa: Folosita pentru transferul de date intre ActivityDataFieldsActivity si MakeProgramActivity
* Nu contine functii
* ContactActivity:
* Sumar clasa: Folosita pentru a contacta CodingShadows prin diferite mijloace
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* sendMail(): trimite un email dupa o interactiune cu utilizatorul
* showWebsite(): trimite utilizatorul pe pagina CodingShadows
* showYouTube(): trimite utilzatorul pe pagina de YouTube a CodingShadows
* showTwitter(): trimite utilzatorul pe pagina de Twitter a CodingShadows
* showFacebook(): trimite utilzatorul pe pagina de Facebook a CodingShadows
* DocumentationActivity (NEFOLOSITA):
* Sumar clasa: Folosita pentru a arata documentatia aplicatiei (NEFOLOSITA)
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* LoadingScreenActivity:
* Sumar clasa: Primul ecran care intampina utilizatorul, logheaza utilizatorul daca este cazul, daca nu il trimite in ecranele corespunzatoare
* onCreate(@Bundle) afiseaza interfata si initializeaza alte functii si elemente din interfata
* onRequestPermissionResult(@RequestCode, @Permissions[], @GrantResults[]): functia este apelata dupa ce utilizatorul accepta permisiunile cerute de aplicatie
* checkStorage(): verifica daca aplicatia are permisiuni de citire/scriere
* checkPrivacyPol(): verifica daca utlizatorul a acceptat Politica de Confidentialitate
* readFileAsString(@fileName): citeste date dintr-un fisier
* checkCrack(): verifica daca aplicatia a fost alterata si o poate invalida
* checkForExisting(): verifica daca se poate loga automat utilizatorul
* MainMenuActivity:
* Sumar clasa: Primul ecran care intampina utilizatorul, logheaza utilizatorul daca este cazul, daca nu il trimite in ecranele corespunzatoare
* onCreate(@Bundle) afiseaza interfata si initializeaza alte functii si elemente din interfata
* showCalendar(): afiseaza calendarul pentru ca utilizatorul sa selecteze o zi
* operateMenu(): afiseaza meniul lateral si seteaza animatia acestuia
* drawLine(): modifica interfata, adaugand o linie orizontala alba
* drawTitle(@title): modifica interfata, adaugand un element tip text
* drawTime(@time): modifica interfata, adaugand un element tip text
* drawDetails(@details): modifica interfata, adaugand un element tip text
* drawComplete(@docID): modifica interfata, adaugand un element tip Switch cu un Tag specific. Cu ajutorul acestui Tag se pot sterge activitati, cand se apasa pe Switch
* drawViews(@title,@timePeriod@details@docID): apeleaza functiile drawLine(), drawTitle(@title), drawTime(@time), drawDetails(@details), drawComplete(@docID), atunci cand se doreste afisarea unei noi activitati
* onClick(@v): (@Override) folosita doar pentru apelarea functiei deleteDoc(@docID) pentru a sterge o activitate atunci cand utilizatorul doreste acest lucru
* deleteDoc(@docID): sterge o activitate atunci cand utilizatorul doreste acest lucru
* getActivitiesData(@dayID): preia date de pe server legate de performata utilizatorului
* updatePerformance(@dayID): actualizeaza date pe server legate de performana utilizatorului
* getUID(): returneaza ID-ul utilizatorului logat
* getActivities(final @date): preia de pe server programul utilizatorului (cu ajutorul altor functii)
* copyPerformanceDoc(final @date, final @dayOfTheWeek): migreaza (server-server) prin intermediul unei copii date legate de performanta utilizatorului pe server
* copyPresetToDate(final @date, final @dayOfTheWeek): creeaza un ID unic cu programul presetat utilizatorului
* createDocument(@date, @title, @time, @details): ajuta la migrarea activitatilor individuale (copyPresetToDate(...))
* showActivities(): afiseaza programul utilizatorului
* sortActivities(): (THREADING - SortActivitiesAsyncTask) sorteaza programul utilizatorului dupa ora de start
* getQuote(): (THREADING - QuoteOfDayAsyncTask) obtine de pe server citatul motivational
* 2 clase : QuoteOfDayAsyncTask, SortActivitiesAsyncTask
* MakeProgramActivity:
* Sumar clasa: Afiseaza performanta utilizatorului pentru o zi selectata de acesta
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* getUID(): returneaza ID-ul utilizatorului logat
* getData(@dayID): preia datele corespunzatoare programului utilizatorului de pe server
* showData(): datele obtinute prin functia getData(@dayID)
* showCalendar(): afiseaza calendarul pentru ca utilizatorul sa selecteze o zi/preset
* MenuActivity:
* Sumar clasa: Afiseaza meniul lateral din care utilizatorul poate deschide activitati
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* finish(): (@Override): seteaza animatia, atunci cand este inchis meniul
* showTutorial(): afiseaza ecranul cu Manualul Utilizatorului
* showPerformance(): afiseaza ecranul de Performanta
* showMakeProgram(): afiseaza configuratorul de activitati
* showContact(): afiseaza ecranul de contact
* showProfile(): afiseaza ecranul cu profilul utilizatorului
* showFriendProfile(): afiseaza ecranul de selectie a prietenului
* showSettings(): afiseaza setarile aplicatiei
* showStayFocused(): afiseaza ecranul de concentrare
* showShop(): afiseaza ecranul cu Magazinul de Relicve
* PerformanceActivity:
* Sumar clasa: Afiseaza ecranul cu performanta utilizatorului
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* getUID(): returneaza ID-ul utilizatorului logat
* getData(@dayID): preia datele corespunzatoare performantei utilizatorului de pe server
* showData(): afiseaza (+ „animatie”) datele obtinute prin functia getData(@dayID)
* showCalendar(): afiseaza calendarul pentru ca utilizatorul sa selecteze o zi
* PermissionDetailsActivity:
* Sumar clasa: Informeaza utilizatorul in legatura cu permisiunile cerute de aplicatie
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* PrivacyPolicyActivity:
* Sumar clasa: Informeaza utilizatorul in legatura cu Politica de Confidentialitate
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* writeStringAsFile(final @fileContents, @fileName): scrie intr-un fisier text selectia utilizatorului
* PrivacyPolicyNoButtonsActivity:
* Sumar clasa: Informeaza utilizatorul in legatura cu Politica de Confidentialitate (fara butoane – special pentru acces din meniul de setari)
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* RegisterActivity:
* Sumar clasa: Inregistrarea utilizatorului
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* register(): inregistreaza utilizatorul
* pushUserData(@userID, @email, @username): pune datele utilizatorului pe server
* SettingsActivity:
* Sumar clasa: Meniu prin intermediul caruia utilizatorul poate accesa setarile aplicatiei sau obtine informatii despre aceasta
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* documentation(): arata documentatia aplicatiei (INACTIVA)
* permission(): arata permisiunile folosite de aplicatie
* showPrivacyPolicy(): arata Politica de Confidentialitate
* logout(): delogheaza utilizatorul
* deleteAcctount(): sterge contul utilizatorului
* onBackPressed(): (@Override): redeschide meniul aplicatiei
* ShopActivity:
* Sumar clasa: Ecran din care utilizatorul poate vedea ce Relicve detine sau poate cumpara Relicve
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* pushData(): actualizeaza datele utilizatorului de pe server
* getData(): obtine datele utilizatorului de pe server
* getUID(): returneaza ID-ul utilizatorului logat
* showAll(): afiseaza Relicvele corespunzator cu datele utilizatorului
* hideAll(): ascunde toate controalele
* SliderAdapter:
* Sumar clasa: Folosita pentru operarea ecranului de ajutor pentru utilizatori
* (static) getSize(): returneaza marimea tutorialului
* getCount(): (@Override) returneaza marimea tutorialului
* isViewFromObject(@view, @object): returneaza ecranul activ
* instantiateItem(@container,@position): (@Override) adauga datele in ecran
* destroyItem(@container, @position, @object): distruge obiectul
* StayFocusedActivity:
* Sumar clasa: Ecran prin intermediul caruia utilizatorul activeaza functia de concentrare
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* onResume(): (@Override)
* onPause(): (@Override) porneste ciclul de verificare a aplicatiilor deschise
* operateMenu(): deschide meniul lateral
* isAccessGranted(): verifica daca aplicatia are permisiunile necesare
* getUsage(): verifica daca exista si care este aplicatia deschisa
* showWhyFail(@reason): ii arata utilizatorului de ce cronometrul s-a oprit
* startStopTimer(): porneste/opreste cronometrul
* stopTimerNoRemember(): opreste cronometrul fara a putea fi repornit
* calcCoins(): calculeaza cate Focus Points trebuie sa primeasca utilizatorul
* addCoins(): porneste ciclul de actualizare a Focus Points
* pushCoins(@coins): actualizeaza moneda utilizatorului pe server
* stopTimer(): opreste cronometrul
* restartTimer(): reporneste cronometrul
* animateCircle(): afiseaza (+animatie) progresul utilizatorului
* checkForAchievements(): verifica daca utilizatorul este eligibil pentru un Achievement
* showFocusPoints(@focusPoints): ii arata utilizatorului cate Focus Points a primit
* hideFocusPoints(): ascunde ceea ce a aratat showFocusPoints(...)
* getUserAchievements(): preia de pe server realizarile utilizatorului
* showAchievementsBadges(@achievmentName): ii ofera utilizatorului un badge pentru anumite realizari
* getUID(): returneaza ID-ul utilizatorului logat
* getHighestTime(@newHigh): preia de pe server timpul maxim de concentrare al utilizatorului
* updateHighestTime(): actualizeaza pe server timpul maxim de concentrare al utilizatorului
* showNotificaionInBar(@title, @text): arata o notificare in bara
* createNotifChannel(): creeaza canalul pentru notificari

* TutorialActivity:
* Sumar clasa: Ecran prin intermediul caruia se afiseaza tutorialul
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* addDotsIndicator(@position): afiseaza parcursul utilizatorului
* UserFriendProfileActivity:
* Sumar clasa: Ecran prin intermediul caruia utilizatorul poate vedea profilul unui prieten
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* getEmailId(): obtine adresa de email a prietenului si apeleaza UserProfileActivity
* UserProfileActivity:
* Sumar clasa: Ecran prin intermediul caruia utilizatorul poate vedea un profil
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata
* getData(): obtine datele necesare de pe server
* displayFocusPoints(): afiseaza Relicvele utilizatorului
* hideAll(): ascunde toate controalele
* showAll(): afiseaza controalele
* showHighestTime(): afiseaza timpul maxim de concentrare
* getUID(): returneaza ID-ul utilizatorului logat
* hideVerifiedBadge(): ascunde insigna de utilizator verificat
* showBadge(): afiseaza insigna (nu aceea de utilizator verificat) utilizatorului
* showVerifiedBadge: afiseaza insigna de utilizator verificat
* animateFocusPoints(): animeaza Focus Points
* VerifyEmailActivity:
* Sumar clasa: Ecran prin intermediul caruia utilizatorului i se cere sa isi verifice adresa de email
* onCreate(@Bundle): afiseaza interfata si initializeaza alte functii si elemente din interfata

Documentatia utilizatorului (manual de utilizare)| Focus: Time Keeper

Un manual de utilizare scurt poate fi gasit in aplicatie: apasati pe butonul din stanga sus, asftel se deschide meniul. Din meniu selectati butonul pe care scrie „Tutorial”. Glisati stanga/dreapta pentru a citi instructiunile. Apasati pe butonul din dreapta jos pe care scrie „Inchide”, pentru a inchide fereastra.

Care este scopul aplicatiei? : Cu ajutorul acestei aplicatii, timpul ajunge din nou in mainile stapanului sau de drept. In mainile tale! In medie, un om pierde pe zi aproximativ 3 ore (conform Inc.com). Speram sa te ajutam sa folosesti acest timp in beneficiul tau!

Utilizari de baza:

* Cum deschid meniul?: se apasa pe butonul cu 3 linii orizontale din partea stanga sus.
* Cum imi modific un program?: se deschide meniul, se apasa pe butonul „Program nou”, se selecteaza ziua/data dorita si apoi se apasa pe + pentru a adauga o activitate noua sau se apasa pe „Sterge toate activitatile din preset” pentru a sterge toate activitatile. Activitatile nu pot fi sterse individual.
* Cum activez modul de concentrare: din meniu, se apasa pe butonul „Stai concentrat”. In fereastra noua, se apasa pe cronometru (00:00) pentru a porni modul.
* Cum pot sa imi vad performanta pentru o anumita zi?: din meniu, se apasa pe butonul „Performanta”, apoi in noul ecran se apasa pe data din partea de sus si se selecteaza ziua dorita.
* Cum pot sa vad profilul unui prieten?: din meniu, se apasa pe „Prieteni” si apoi se introduce adresa de email a prietenului.

Probleme care pot aparea in momentul utilizarii:

* Activitatile mele nu sunt afisate: va rugam sa verificati conexiunea dvs. la internet. Daca aveti o conexiune mai slaba la internet, activitatile pot aparea ceva mai tarziu. Daca aveti o conexiune buna la internet si activitatile nu apar in maxim 3 secunde, reselectati data, apasand pe butonul din partea de sus a ecranului. Daca tot nu apar activitatile si sunteti sigur ca aveti un program introdus pentru acea zi, va rugam sa ne contactati (Email: [contact@codingshadows.com](mailto:contact@codingshadows.com) | Facebook: @CodingShadows).
* Performanta mea nu este afisata: performanta este afisata doar pentru zilele in care ati avut si completat activitati. Daca in acest caz tot nu apar datele, va rugam sa reselectati data, apasand pe butonul din partea de sus a ecranului. Daca problema persista, va rugam sa ne contactati (Email: [contact@codingshadows.com](mailto:contact@codingshadows.com) | Facebook: @CodingShadows).
* Cronometrul din ecranul „Stai concentrat” se opreste automat desi nu am folosit nici o alta aplicatie: aceasta eroare poate aparea din cauza faptului ca o aplicatie care rula in fundal a fost detectata ca fiind deschisa. Aceasta eroare este foarte rar intalnita. Va rugam sa reporniti aplicaita (Focus: Time Keeper), iar daca problema persista va rugam sa reporniti telefonul.