

LATVIJAS UNIVERSITĀTE

DATORIKAS FAKULTĀTE

Android lietotne – uzlabots modinātājs

KVALIFIKĀCIJAS DARBS

Autors: Andris Niedrītis

Stud. apl. nr: an10031

Darba vadītājs: Gundars Lisovskis­­

RĪGA 2014

# Anotācija

Android lietotne – uzlabots modinātājs ir Android viedtālruņu lietotājiem paredzēta lietojumprogramma, kuri nav apmierināti ar telefonā jau esošo modinātāja lietojumprogrammatūru.

Programma piedāvā tās pašas funkcijas, kādas parasti jau ir viedtālruņos esošajos modinātājos, kā arī papildus funkcionalitāti, kāda varētu vēl noderēt. Piemēram, izvēlēties citādāku atkārtošanas intervālu, konkrētas dienas katru nedēļu, vai tikai tad, kad tas ir aktivizēts, kā arī īpaši grūti pamodināmiem cilvēkiem dot iespēju uzstādīt īpašu uzdevumu, kurš ir jāveic, lai varētu izslēgt modinātāju, tādējādi labāk pamodinot šo cilvēku.

Atslēgas vārdi: Android, Modinātājs, Modinātāja uzdevums

# Abstract

Android application – improved alarm clock is application for Android smartphone users, who are not satisfied with alarm clock, that is allready existing in their phones.

This program offers the same functionality, that is allready in these smartphones aswell as some added functionality, which might be usefull. For example, it is possible to chose different repeating interval, exact days every week, or only then, when it is active, and also for people, that are really hard to get waked up, there would be an option to set a special task, that must be completed in order to turn of the alarm. That way these people would wake up easier.

Key words: Android, Alarm clock, Task of alarm clock

# Saturs

[Anotācija 2](#_Toc389449736)

[Abstract 3](#_Toc389449737)

[Saturs 4](#_Toc389449738)

[Ievads 7](#_Toc389449739)

[PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA 8](#_Toc389449740)

[1. Ievads 8](#_Toc389449741)

[1.1. Nolūks 8](#_Toc389449742)

[1.2. Darbības sfēra 8](#_Toc389449743)

[1.3. Definīcijas un saīsinājumi 8](#_Toc389449744)

[1.4. Saistība ar citiem dokumentiem 9](#_Toc389449745)

[1.5. Dokumenta pārskats 10](#_Toc389449746)

[2. Vispārējs apraksts 11](#_Toc389449747)

[2.1. Produkta perspektīva 11](#_Toc389449748)

[2.2. Biznesa prasības 11](#_Toc389449749)

[2.3. Sistēmas lietotāju grupas 11](#_Toc389449750)

[2.4. Lietotāja raksturiezīmes 11](#_Toc389449751)

[2.5. Produkta funkcijas 11](#_Toc389449752)

[2.6. Vispārēji ierobežojumi 12](#_Toc389449753)

[2.7. Pieņēmumi un atkarības 12](#_Toc389449754)

[3. Funkcionālās prasības 13](#_Toc389449755)

[3.1. Rādīt sākumlapu – visu modinātāju sarakstu 13](#_Toc389449756)

[3.2. Atvērt modinātāja pievienošanas/rediģēšanas skatu 13](#_Toc389449757)

[3.3. Izvēlēties laiku 14](#_Toc389449758)

[3.4. Ievadīt modinātāja nosaukumu 14](#_Toc389449759)

[3.5. Izvēlēties atkārtošanas dienas 14](#_Toc389449760)

[3.6. Izvēlēties iknedēļas atkārtojumu 15](#_Toc389449761)

[3.7. Izvēlēties modinātāja signālu 15](#_Toc389449762)

[3.8. Izvēlēties izpildāmā uzdevuma tipu 16](#_Toc389449763)

[3.9. Izvēlēties uzdevuma sarežģītības pakāpi 16](#_Toc389449764)

[3.10. Pievienot/Saglabāt modinātāju 17](#_Toc389449765)

[3.11. Atvērt modinātāja rediģēšanas skatu 17](#_Toc389449766)

[3.12. Rediģēt modinātāja nosaukumu 17](#_Toc389449767)

[3.13. Rediģēt atkārtošanas dienas 18](#_Toc389449768)

[3.14. Rediģēt iknedēļas atkārtojumu 18](#_Toc389449769)

[3.15. Rediģēt modinātāja signālu 19](#_Toc389449770)

[3.16. Rediģēt izpildāmā uzdevuma tipu 19](#_Toc389449771)

[3.17. Rediģēt uzdevuma sarežģītības pakāpi 20](#_Toc389449772)

[3.18. Dzēst modinātāju 20](#_Toc389449773)

[3.19. Apstiprināt/Atcelt modinātāja dzēšanu 21](#_Toc389449774)

[3.21. Izpildīt uzdevumu 21](#_Toc389449775)

[3.22. Izslēgt modinātāju 22](#_Toc389449776)

[3.23. Aktivizēt modinātāju 22](#_Toc389449777)

[3.24. Deaktivizēt modinātāju 23](#_Toc389449778)

[4. Saskarņu saraksts 24](#_Toc389449779)

[4.1. Lietotāja saskarne 24](#_Toc389449780)

[4.2. Aparatūras saskarne 25](#_Toc389449781)

[4.3. Programmatūras saskarne 25](#_Toc389449782)

[5. Nefunkcionālās prasības 26](#_Toc389449783)

[5.1. Veiktspējas prasības 26](#_Toc389449784)

[5.2. Pieejamība 26](#_Toc389449785)

[5.3. Lietojamība 26](#_Toc389449786)

[5.4. Datu drošība 26](#_Toc389449787)

[5.5. Uzticamība 26](#_Toc389449788)

[PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS 27](#_Toc389449789)

[1. Ievads 27](#_Toc389449790)

[1.1. Nolūks 27](#_Toc389449791)

[1.2. Darbības sfēra 27](#_Toc389449792)

[1.3. Definīcijas 27](#_Toc389449793)

[1.4. Saistība ar citiem dokumentiem 28](#_Toc389449794)

[2. Dekompozīcijas apraksts 28](#_Toc389449795)

[2.1. Moduļu dekompozīcija 28](#_Toc389449796)

[2.2. Datu dekompozīcija 30](#_Toc389449797)

[3. Atkarības apraksts 32](#_Toc389449798)

[3.1. Starpmoduļu atkarības 32](#_Toc389449799)

[3.2. Datu atkarības 33](#_Toc389449800)

[4. Detalizēts projektējums 34](#_Toc389449801)

[4.1. Datu detalizēts projektējums 34](#_Toc389449802)

[4.1.1. Tabula „alarms” 34](#_Toc389449803)

[4.1.2. Tabula „counting\_tasks\_arabicnr” 34](#_Toc389449804)

[4.1.3. Tabula „multiply\_tasks\_arabicnr” 35](#_Toc389449805)

[4.1.4. Tabula substraction\_tasks\_arabicnr 35](#_Toc389449806)

[4.1.5. Tabula dividing\_tasks\_arabicnr 35](#_Toc389449807)

[4.1.6. Tabula counting\_tasks\_romannr 36](#_Toc389449808)

[4.1.7. Tabula multiply\_tasks\_romannr 36](#_Toc389449809)

[4.1.8. Tabula substraction\_tasks\_romannr 36](#_Toc389449810)

[4.1.9. Tabula dividing\_tasks\_romannr 37](#_Toc389449811)

[TESTĒŠANAS DOKUMENTĀCIJA 38](#_Toc389449812)

[Ievads 38](#_Toc389449813)

[Testēšanas rezultāti 38](#_Toc389449814)

[EKRĀNFORMU PIEMĒRI 44](#_Toc389449815)

[1. Sākuma skats 44](#_Toc389449816)

[2. Modinātāja pievienošanas/rediģēšanas skats 45](#_Toc389449817)

[3. Modinātājs saglabāts 46](#_Toc389449818)

[4. Modinātāja aktivizēšanās skats – modināšana. 47](#_Toc389449819)

[5. Modinātāja dzēšana 48](#_Toc389449820)

[PROJEKTA ORGANIZĀCIJA 49](#_Toc389449821)

[KVALITĀTES NODROŠINĀŠANA 50](#_Toc389449822)

[KONFIGURĀCIJU PARVALDĪBA 51](#_Toc389449823)

[DARBIETILPĪBAS NOVĒRTĒJUMS 52](#_Toc389449824)

[IZMANTOTĀS TEHNOLOĢIJAS 54](#_Toc389449825)

[SECINĀJUMI 55](#_Toc389449826)

[IZMANTOTĀ LITERATŪRA 56](#_Toc389449827)

[PIELIKUMI 57](#_Toc389449828)

[PROGRAMMATŪRAS PIRMKODS 57](#_Toc389449829)

# Ievads

Ar katru dienu viedo tālruņu lietotāju skaits palielinās, un šī tendence labi parāda mūsdienu pasaules cilvēka vēlmes – ātri, viegli, kompakti. Neko nevar darīt ātri, viegli un kompakti, ja rīts sākas jau kavējot, tāpēc ir nepieciešams pamosties laikā ar pirmo modinātāja zvanu nevis atlikt vairākas reizes un pašā pēdējā brīdī doties dienas gaitās. Domājot par to kā labāk iesākt dienas ritmo radās ideja uzlabot vienu no šī brīža svarīgākajām tālruņa funkcijām, kuru ik dienas lieto neskaitāmi daudz cilvēku.

Darba mērķis bija izstrādāt "Androd lietojumprogramma – uzlabots modinātājs", kas viņas lietotājiem sniegtu iespēju uzsākt dienu laikā, ko viņi ir izvēlējušies uzstādot modinātāju, nevis pārdomājot vairākas reizes no rīta, kad modinātājs atskan. Programma liktu izpildīt uzdevumu pirms modinātājs apklustu, kas nodrošinātu to, ka cilvēks tiešām pamostos nevis pat nepaskatoties uz telefonu izslēgtu modinātāju. Lietotājs varētu izvēlēties vienu no vairākiem uzdevumu veidiem, un vienu no vairākām sarežģītības pakāpēm. Izstrādātā sistēma ir Android lietojumprogrammatūra, kura ir paredzēta Android viedtālruņiem. Sistēma tika izstrādāta tā, lai tās saskarne būtu pēc iespējas vienkāršāka un viegli izmantojama.

Kvalifikācijas darba ietvaros tika izstrādāta programmatūras prasību specifikācijas, projektējuma apraksta, testēšanas dokumentācijas, projekta organizācijas, kvalitātes nodrošināšanas, konfigurāciju pārvaldības, dots darbietilpības novērtējuma, aprakstītas izmantotas tehnoloģijas un veikti secinājumi par kvalifikācijas darbā paveikto. Pievienots arī programmatūras pirmkods un pielikumi.

# PROGRAMMATŪRAS PRASĪBU SPECIFIKĀCIJA

## 1. Ievads

### 1.1. Nolūks

Dokumenta nolūks ir apkopot un aprakstīt sistēmas „Android lietotne – uzlabots modinātājs” prasības – gan funkcionālās, gan nefunkcionālās.

Balstoties uz šo dokumentu turpmāk tiks izstrādāta cita nepieciešamā dokumentācija, kā arī pati sistēma.

### 1.2. Darbības sfēra

Šī lietojumprogramma ir paredzēta lietotājiem, kurus neapmierina iebūvētā modinātāja funkcionalitāte. Viņiem būs iespēja izvēlēties dažādus uzstādījumus, un pielāgot programmu savām vajadzībām, lai tā noderētu pēc iespējas labāk.

### 1.3. Definīcijas un saīsinājumi

#### 1.3.1. Jēdzieni

|  |  |
| --- | --- |
| **Jēdziens** | **Skaidrojums** |
| Android viedtālrunis | Mobilā telefona aparāts, kas nodrošina ne tikai balss pakalpojumu, bet arī e-pasta un teksta ziņojumu sūtīšanu, piekļuvi tīmeklim un balss atpazīšanu. |
| Pieskarties | Pirksta nolaišana uz ekrāna. |
| Skārienekrāns | Displeja ekrāns, kas pārklāts ar caurspīdīgu skārienjūtīgu pārklājumu. Skārienekrāns pārveido lietotāja pirksta pieskārienu kādā noteiktā ekrāna vietā informācijā, kas tiek nodota programmatūrai. Tas dod iespēju skārienekrānu izmantot peles, gaismas zīmuļa vai tml. rādītājierīču aizstāšanai. |
| Lietotājs | Persona, kas izmanto lietojumprogrammu. |
| Datubāze | Savstarpēji saistītu informacionālu objektu tematisks kopums, kas ar speciālas pārvaldības sistēmas starpniecību organizēts tā, lai nodrošinātu ērtu informācijas izguvi, izdarītu tās atlasi un kārtošanu. Informācija datu bāzē parasti ir sadalīta ierakstos (tabulās), no kuriem katram var būt viens vai vairāki lauki (kolonnas). |
| UTF-8 | Mainīga platuma kodējums. Tas var attēlot jebkuru simbolu unikoda standartā, un vienlaikus ir savietojams ar ASCII. Tas ir guvis plašu izplatību e-pastā, globālajā tīmeklī un citos teksta glabāšanas vai pārsūtīšanas lietojumos. |
| URI | Saite, kas norāda uz skaņas failu telefonā. |
| Darbību josla | Augšējā josla programmas skatā, kurā var atrasties kāda darbības poga, kā arī programmas logo. (angļu valodā, Android termins - Action bar) |

#### 1.3.2. Pieņemtie datu tipi

|  |  |
| --- | --- |
| **Datu tips** | **Skaidrojums** |
| Izvēle no saraksta | Viens no saraksta elementiem. Tiek izmantots, kad lietotājam vajag veikt izvēli no vairākiem elementiem. Datus iegūst, kad lietotājs pieskaras kādam no saraksta elementiem. |
| Teksts | UTF-8 simbolu virkne. |

### 1.4. Saistība ar citiem dokumentiem

Šis dokuments tika izstrādāts pēc LVS 68:1996 "Programmatūras Prasību Specifikācijas Ceļvedis" un ir lietojams kopā ar sekojošiem dokumentiem - „Programmatūras projektējuma apraksts”, „Testēšanas dokumentācija”, „Konfigurāciju pārvaldība”, „Kvalitātes nodrošināšana”, ‖Projekta organizācija”, „Darbietilpības novērtējums”.

### 1.5. Dokumenta pārskats

Dokuments sastāv no 5 nodaļām: ievads, vispārējs apraksts, funkcionālās prasības, ārējā saskarne, nefunkcionālās prasības.

Ievadā aprakstīti dokumenta mērķi, nolūks, izmantotie jēdzieni, kas sastopami dokumentā.

Vispārējais apraksts tiek sniegts pārskats par funkcijām, kas sistēmai ir jāpilda, kā arī aprakstītas lietotāja raksturiezīmes. Šajā nodaļā ir aprakstīti ierobeţojumi un atkarības, kas var ietekmēt sistēmas izstrādi.

Funkcionālo prasību nodaļā tiek aprakstītas sistēmā iekļautās funkcijas, to ieejas dati, šo datu apstrādes darbības, izejas dati, funkciju atkarības.

## 2. Vispārējs apraksts

### 2.1. Produkta perspektīva

Produkts ir neatkarīgs un pilnīgi pašpietiekams, tā darbība nav atkarīga no citu sistēmu darbības.

### 2.2. Biznesa prasības

|  |  |
| --- | --- |
| **Prasība** | **Identifikators** |
| Jādod lietotājam iespēja pievienot, rediģēt un dzēst modinātāju. | B.1 |
| Jādod lietotājam iespēja izvēlēties uzdevumu un tā sarežģītības pakāpi konkrētam modinātājam. | B.2 |

### 2.3. Sistēmas lietotāju grupas

Sistēmas lietošanā ir paredzēts viena sistēmas lietotāju grupa – lietotājs. Lietotājs ir cilvēks, kurš izmanto mobilo lietojumprogrammu un var darboties ar visām tās funkcijām bez ierobežojumiem.

### 2.4. Lietotāja raksturiezīmes

Lietotājs, ir cilvēks, kurš izmanto mobilas lietojumprogrammatūras, un kurām ir pieredze darbā ar tām.

### 2.5. Produkta funkcijas

Katrai funkcijai ir savs identifikācijas numurs. Produkta funkcijas:

1. Rādīt sākumlapu – visu modinātāju sarakstu
2. Atvērt modinātāja pievienošanas/rediģēšanas skatu
3. Izvēlēties laiku
4. Ievadīt modinātāja nosaukumu
5. Izvēlēties atkārtošanas dienas
6. Izvēlēties iknedēļas atkārtojumu
7. Izvēlēties modinātāja signālu
8. Izvēlēties izpildāmā uzdevuma tipu
9. Izvēlēties uzdevuma sarežģītības pakāpi
10. Pievienot/Saglabāt modinātāju
11. Rediģēt modinātāju
12. Rediģēt modinātāja nosaukumu
13. Rediģēt atkārtošanas dienas
14. Rediģēt iknedēļas atkārtojumu
15. Rediģēt modinātāja signālu
16. Rediģēt izpildāmā uzdevuma tipu
17. Rediģēt uzdevuma sarežģītības pakāpi
18. Dzēst modinātāju
19. Apstiprināt/Atcelt modinātāja dzēšanu
20. Atvērt pamatuzstādījumus
21. Izpildīt uzdevumu
22. Izslēgt modinātāju
23. Aktivizēt modinātāju
24. Deaktivizēt modinātāju

### 2.6. Vispārēji ierobežojumi

Sistēmas lietošanai nepieciešama specifiska iekārta – viedais tālrunis ( smart phone ) ar Android 3.0 vai jaunāku operētājsistēmas versiju.

### 2.7. Pieņēmumi un atkarības

Šī sistēma ir pieejama lietotājiem ar Android viedtālruņiem, kuru operētājsistēmas versija ir 3.0 vai jaunāka.

Sistēmas darbības korektums atkarīga no lietotāja saglabātajām izvēlēm un datu korektums atkarīgs tikai un vienīgi no lietotāja darbībām, nav citu sistēmu iedarbība.

Sistēmas darbība ir atkarīga no ierīces darbības, piemēram, ja ierīces skaļrunis ir bojāts, sistēmas funkcionalitāte būs nepilnīga.

## 3. Funkcionālās prasības

### 3.1. Rādīt sākumlapu – visu modinātāju sarakstu

|  |  |
| --- | --- |
| ID | 1 |
| Mērķis | Attēlot programmas pamatskatu, attēlot visu modinātāju sarakstu, ja tādi ir. |
| Dod pieeju pie | 2: Atvērt modinātāja pievienošanas skatu, 11: Rediģēt modinātāju 19:Dzēst modinātāju, 24:Aktivizēt modinātāju, 25:Deaktivizēt modinātāju |
| Ievaddati | |
| - | |
| Apstrāde | |
| - | |
| Izvaddati | |
| Tiek attēlots galvenais skats. Ja ir saglabāti modinātāji, tiek attēlots modinātāju saraksts, virs modinātāju saraksta atrodas darbību josla, kurā ir poga ar kuru var pievienot jaunu modinātāju – 2: Atvērt modinātāja pievienošanas skatu. | |

### 3.2. Atvērt modinātāja pievienošanas/rediģēšanas skatu

|  |  |
| --- | --- |
| ID | 2 |
| Mērķis | Attēlot skatu, kurā var izvēlēties informāciju par modinātāju un to saglabāt. |
| Dod pieeju pie | - |
| Ievaddati | |
| Nospiesta poga „+” sākuma skatā. Poga atrodas darbību joslā. | |
| Apstrāde | |
| Tiek atvērts skats, kurā redzami visi nepieciešamie lauki modinātāja saglabāšanai. | |
| Izvaddati | |
| - | |

### 3.3. Izvēlēties laiku

|  |  |
| --- | --- |
| ID | 3 |
| Mērķis | Izvēlēties laiku, kurā modinātājs sāks darboties. |
| Dod pieeju pie | - |
| Ievaddati | |
| Stundas, minūtes. | |
| Apstrāde | |
| Ievada izmantojot Android laika izvēlēšanās ritjoslu. Izvēlēto laiku saglabā datubāzē un servisā, kuru izveido saglabājot modinātāju. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Izvēlētais laiks tiek attēlots konkrētajā ierīcē. | |

### 3.4. Ievadīt modinātāja nosaukumu

|  |  |
| --- | --- |
| ID | 4 |
| Mērķis | Ievadīt modinātāja nosaukumu, lai varētu tos atšķirt. |
| Dod pieeju pie | - |
| Ievaddati | |
| Teksts, UTF-8 kodējumā. | |
| Apstrāde | |
| Ievada tam paredzētā laukā. Tekstu saglabā datubāzē un servisā, kuru izveido saglabājot modinātāju. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Ievadītais teksts rādās konkrētajā lauciņā. | |

### 3.5. Izvēlēties atkārtošanas dienas

|  |  |
| --- | --- |
| ID | 5 |
| Mērķis | Izvēlēties dienas, kurās modinātājs veiks modināšanas darbību. |
| Dod pieeju pie | - |
| Ievaddati | |
| Uzspiest uz pogas „ieslēgts/izslēgts”. Atkārtošanas dienu saraksts: Svētdiena, Pirmdiena, Otrdiena, Trešdiena, Ceturtdiena, Piektdiena, Sestdiena. | |
| Apstrāde | |
| Katras dienas konkrētā izvēle tiek saglabāta datubāzē un servisā. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Statusa ”ieslēgts/izslēgts” izmaiņas tiek attēlotas, kad poga tiek nospiesta. | |

### 3.6. Izvēlēties iknedēļas atkārtojumu

|  |  |
| --- | --- |
| ID | 6 |
| Mērķis | Izvēlēties vai atkārtot modinātāju katru nedēļu, vai tikai tad, kad tas ir aktivizēts vēlreiz kādā citā programmas darbības nedēļā. |
| Dod pieeju pie | - |
| Ievaddati | |
| Uzspiest uz pogas „ieslēgts/izslēgts”. | |
| Apstrāde | |
| Izvēle tiek saglabāta datubāzē un servisā. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Statusa ”ieslēgts/izslēgts” izmaiņas tiek attēlotas, kad poga tiek nospiesta. | |

### 3.7. Izvēlēties modinātāja signālu

|  |  |
| --- | --- |
| ID | 7 |
| Mērķis | Izvēlēties kāds signāls atskanēs modinātājam sākot darboties. |
| Dod pieeju pie | - |
| Ievaddati | |
| Izvēle no saraksta. | |
| Apstrāde | |
| Saraksts tiek ņemts no telefonā jau esošajām modinātāja skaņām. Datubāzē un servisā, kuru izveido saglabājot modinātāju, saglabā konkrētā skaņas signāla URI. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Izvēle redzama konkrētajā lauciņā. | |

### 3.8. Izvēlēties izpildāmā uzdevuma tipu

|  |  |
| --- | --- |
| ID | 8 |
| Mērķis | Izvēlēties kāda tipa uzdevums būs jāizpilda, lai izslēgtu modinātāju, kad tas sāktu darbību. |
| Dod pieeju pie | - |
| Ievaddati | |
| Izvēle no saraksta. Sarakstā atrodas šādi tipi: Saskaitīšana, ar arābu cipariem; Atņemšana, ar arābu cipariem; Reizināšana, ar arābu cipariem; Dalīšana, ar arābu cipariem; Saskaitīšana, ar romiešu cipariem; Atņemšana, ar romiešu cipariem; Reizināšana, ar romiešu cipariem; Dalīšana, ar romiešu cipariem | |
| Apstrāde | |
| Izvēli no saraksta saglabā datu bāzē un servisā, kuru izveido saglabājot modinātāju. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Konkrētajā lauciņā rādās izvēlātais tips. | |

### 3.9. Izvēlēties uzdevuma sarežģītības pakāpi

|  |  |
| --- | --- |
| ID | 9 |
| Mērķis | Izvēlēties kāda sarežģītības pakāpe būs uzdevumam, kuru būs jāpilda, lai izslēgtu modinātāju, kad tas sāktu darbību. |
| Dod pieeju pie | - |
| Ievaddati | |
| Izvēle no saraksta. Sarakstā atrodas šādi tipi: viegls, vidējs, sarežģīts. | |
| Apstrāde | |
| Izvēli no saraksta saglabā datu bāzē un servisā, kuru izveido saglabājot modinātāju. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Konkrētajā lauciņā rādās izvēlātais tips. | |

### 3.10. Pievienot/Saglabāt modinātāju

|  |  |
| --- | --- |
| ID | 10 |
| Mērķis | Pievienot modinātāju sarakstam, padarīt to aktīvu. |
| Dod pieeju pie | 1: Rādīt sākumlapu – visu modinātāju sarakstu |
| Ievaddati | |
| Nospiesta saglabāšanas poga darbību joslā. | |
| Apstrāde | |
| Tiek saglabāti visi skata dati gan datubāzē, gan servisā, kas izsauks modinātāja darbību. | |
| Izvaddati | |
| Tiek atvērts sākuma skats. ( 1: Rādīt sākumlapu – visu modinātāju sarakstu ) | |

### 3.11. Atvērt modinātāja rediģēšanas skatu

|  |  |
| --- | --- |
| ID | 11 |
| Mērķis | Veikt izmaiņas esoša modinātāja uzstādījumos. |
| Atkarība | - |
| Dod pieeju pie | 2: Atvērt modinātāja pievienošanas/ skatu |
| Ievaddati | |
| Pieskāriens pie modinātāja. | |
| Apstrāde | |
| Tiek atvērts modinātāja pievienošanas/rediģēšanas skats, skatā katrā laukā jau tiek attēlota informācija, kas tika saglabāta par konkrēto modinātāju. | |
| Izvaddati | |
| Tiek atvērts modinātāja pievienošanas/rediģēšanas skats | |

### 3.12. Rediģēt modinātāja nosaukumu

|  |  |
| --- | --- |
| ID | 12 |
| Mērķis | Mainīt konkrētā modinātāja nosaukumu |
| Atkarība | - |
| Dod pieeju pie | - |
| Ievaddati | |
| Modinātāja esošais nosaukums. Teksts UTF-8 kodējumā. | |
| Apstrāde | |
| Ievada tam paredzētā laukā. Tekstu saglabā datubāzē un servisā, kuru izveido saglabājot modinātāju. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Mainītais nosaukums redzams konkrētajā lauciņā. | |

### 3.13. Rediģēt atkārtošanas dienas

|  |  |
| --- | --- |
| ID | 13 |
| Mērķis | Mainīt to, kurās dienās modinātājs darbojas. |
| Atkarība | - |
| Dod pieeju pie | - |
| Ievaddati | |
| Esošās atkārtošanas dienas ( Ieslēgts vai izslēgts atkārtojums ). | |
| Apstrāde | |
| Statusa ”ieslēgts/izslēgts” izmaiņas tiek attēlotas, kad poga tiek nospiesta. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Statusa ”ieslēgts/izslēgts” izmaiņas tiek attēlotas, kad poga tiek nospiesta. | |

### 3.14. Rediģēt iknedēļas atkārtojumu

|  |  |
| --- | --- |
| ID | 14 |
| Mērķis | Izvēlēties vai atkārtot modinātāju katru nedēļu, vai tikai tad, kad tas ir aktivizēts vēlreiz kādā citā programmas darbības nedēļā. Mainīt šo stāvokli uz pretējo. |
| Atkarība | - |
| Dod pieeju pie | - |
| Ievaddati | |
| Esošais stāvoklis ( Ieslēgts vai izslēgts atkārtojums ). | |
| Apstrāde | |
| Statusa ”ieslēgts/izslēgts” izmaiņas tiek attēlotas, kad poga tiek nospiesta. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Statusa ”ieslēgts/izslēgts” izmaiņas tiek attēlotas, kad poga tiek nospiesta. | |

### 3.15. Rediģēt modinātāja signālu

|  |  |
| --- | --- |
| ID | 15 |
| Mērķis | Mainīt esošo modinātāja signālu. |
| Atkarība | - |
| Dod pieeju pie | - |
| Ievaddati | |
| Esošā modinātāja signāla URI. | |
| Apstrāde | |
| Esošais modinātāja signāls ir atzīmēts signālu sarakstā. Pēc jaunās izvēles veikšanas tiek piefikēts jaunais signāls un tā URI saite. To saglabā datubāzē un servisā. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Signāla nosaukums konkrētajā lauciņā. | |

### 3.16. Rediģēt izpildāmā uzdevuma tipu

|  |  |
| --- | --- |
| ID | 16 |
| Mērķis | Mainīt izpildāmā uzdevuma tipu. |
| Atkarība | - |
| Dod pieeju pie | - |
| Ievaddati | |
| Esošais uzdevuma tips. | |
| Apstrāde | |
| Atverot sarakstu esošais uzdevuma tips ir atzīmēts kā izvēlēts, lietotājs var izvēlēties jauno un atzīmēt. Izvēli no saraksta saglabā datu bāzē un servisā, kuru izveido saglabājot modinātāju. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Konkrētajā lauciņā rādās izvēlātais tips. | |

### 3.17. Rediģēt uzdevuma sarežģītības pakāpi

|  |  |
| --- | --- |
| ID | 17 |
| Mērķis | Mainīt uzdevuma esošo sarežģītības pakāpi. |
| Atkarība | - |
| Dod pieeju pie | - |
| Ievaddati | |
| Esošā sarežģītības pakāpe. | |
| Apstrāde | |
| Atverot sarakstu esošā sarežģītības pakāpe ir atzīmēta kā izvēlēta, lietotājs var izvēlēties jauno un atzīmēt. Izvēli no saraksta saglabā datu bāzē un servisā, kuru izveido saglabājot modinātāju. Šī darbība notiek, kad tiek izpildīga saglabāšanas darbība (10: Pievienot/Saglabāt modinātāju). | |
| Izvaddati | |
| Konkrētajā lauciņā rādās izvēlātā sarežģītības pakāpe. | |

### 3.18. Dzēst modinātāju

|  |  |
| --- | --- |
| ID | 18 |
| Mērķis | Dzēst kādu no esošajiem modinātājiem. |
| Atkarība | - |
| Dod pieeju pie | 19: Apstiprināt/Atcelt modinātāja dzēšanu |
| Ievaddati | |
| Ilgais pieskāriens pie kāda no modinātājiem. | |
| Apstrāde | |
| Tiek izveidots jauns skats, kurā pārjautā vai lietotājs vēlas dzēst šo modinātāju. | |
| Izvaddati | |
| - | |

### 3.19. Apstiprināt/Atcelt modinātāja dzēšanu

|  |  |
| --- | --- |
| ID | 19 |
| Mērķis | Pārbaudīt, vai lietotājs tiešām vēlas dzēst modinātāju. |
| Atkarība | - |
| Dod pieeju pie | - |
| Ievaddati | |
| Pogas „Ok” vai „Cancel” nospiešana. | |
| Apstrāde | |
| Spiežot „Ok” tiek saņemts modinātāja ID, dzēsts tabulas ieraksts un serviss. Ja lietotājs nevēlas dzēst modinātāju, var spiest „cancel”. Pēc darbības veikšanas tiek atvērts sākuma skats. | |
| Izvaddati | |
| Tiek attēlots sākuma skats – modinātāju saraksts. | |

### 3.21. Izpildīt uzdevumu

|  |  |
| --- | --- |
| ID | 21 |
| Mērķis | Likt lietotājam izpildīt uzdevumu, lai viņš varētu izslēgt modinātāju. |
| Atkarība | - |
| Dod pieeju pie | 22: Izslēgt modinātāju |
| Ievaddati | |
| Modinātāja uzdevums (tiek attēlots uz ekrāna), uzdevuma pareizā atbilde( tiek izmanota pārbaudē – salīdzina ar lietotāja ievadīto). | |
| Apstrāde | |
| Uz ekrāna ir redzams uzdevums un iespēja ailītē ierakstīt pareizo atbildi. Ja lietotājs ieraksta pareizo atbildi, tiek aktivizēta modinātāja izslēgšanas poga, ja nē, tiek nodzēsta atbilde un modinātājs turpina savu darbību. | |
| Izvaddati | |
| - | |

### 3.22. Izslēgt modinātāju

|  |  |
| --- | --- |
| ID | 22 |
| Mērķis | Beigt modināšanas darbību. |
| Atkarība | - |
| Dod pieeju pie | 1: Rādīt sākumlapu – visu modinātāju sarakstu |
| Ievaddati | |
| Ja ir bijis uzdevums, tad apstiprinājums, ka tas ir izpildīts pareizi. ( poga izslēgt aktivizējas, pirms tam ir deaktivizēta ) | |
| Apstrāde | |
| Pēc modinātāja aktivizēšanās tikai poga Dismiss to izslēdz, citas telefona pogas šajā laika momentā nedarbojas. Pēc pogas nospiešanas modinātājs apklust, tas netiek izsaukts atkārtoti pēc kāda laika intervāla. Tiek atvērts sākuma kats pat ja programma iepriekš nebija ieslēgta. | |
| Izvaddati | |
| Tiek izslēgts modinātājs, tiek parādīts sākuma skats – visu modinātāju saraksts. | |

### 3.23. Aktivizēt modinātāju

|  |  |
| --- | --- |
| ID | 23 |
| Mērķis | Padarīt modinātāju aktīvu. |
| Atkarība | - |
| Dod pieeju pie | - |
| Ievaddati | |
| Iepriekš modinātājs bijis deaktivizētā stāvoklī. | |
| Apstrāde | |
| Pēc pogas nospiešanas tiek mainīts pogas stāvoklis „Izslēgts” uz „Ieslēgts”, tiek izmainīta vērtība datubāzē un konkrētā modinātāja servisā. | |
| Izvaddati | |
| Mainās pogas stāvoklis no „Izslēgts” uz „Ieslēgts”. | |

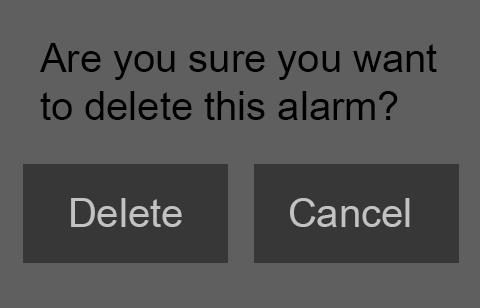
### 3.24. Deaktivizēt modinātāju

|  |  |
| --- | --- |
| ID | 24 |
| Mērķis | Padarīt modinātāju neaktīvu. |
| Atkarība | - |
| Dod pieeju pie | - |
| Ievaddati | |
| Iepriekš modinātājs bijis aktivizētā stāvoklī. | |
| Apstrāde | |
| Pēc pogas nospiešanas tiek mainīts pogas stāvoklis „Ieslēgts” uz „Izslēgts”, tiek izmainīta vērtība datubāzē un konkrētā modinātāja servisā. | |
| Izvaddati | |
| Mainās pogas stāvoklis no „Ieslēgts” uz „Izslēgts”. | |

## 4. Saskarņu saraksts

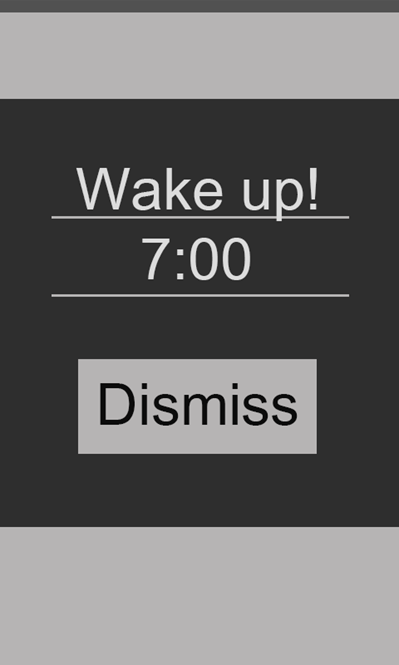
### 4.1. Lietotāja saskarne

Sistēmai jāatbalsta skārienekrānam pieejamos pirkstu žestus, kuri izpilda pogu uzspiešanu un ritināšanu. Programma izmanto paziņojumus lietotājam divos gadījumos – ja lietotājs izvēlas dzēst kādu modinātāju, tiek pārjautāts, vai tiešām lietotājs to vēlas darīt (1) un otrs gadījums ir tad, kad notiek reālā modinātāja darbība(2).



1.Modinātāja dzēšana.

Nospiežot Delete tiks dzēsts izvēlētais modinātājs, nospiežot Cancel, programma atgriezīsies sākuma skatā un nedzēsīs konkrēto modinātāju.

2.Modinātāja aktivizēšanās skats.

Nospiežot Dismiss modinātājs tiks izslēgts.

### 4.2. Aparatūras saskarne

Lai piekļūtu sistēmai un izmantotu tās iespējas, klientam nav nepieciešama papildus aparatūra, viss notiek lietotāja telefonā.

### 4.3. Programmatūras saskarne

Lai piekļūtu sistēmai un izmantotu tās iespējas, klientam nav nepieciešama papildus programmatūra.

## 5. Nefunkcionālās prasības

### 5.1. Veiktspējas prasības

Tā kā sistēmu izmanto viens lietotājs un tā nav saistīta ar ārējiem resursiem, tai nav jāspēj nodrošināt vairāku lietotāju darbību. Sistēmas ātrdarbība ir minimāli atkarīga no mobilās ierīces parametriem un ātrdarbības.

### 5.2. Pieejamība

Sistēmai jābūt pieejamai jebkurā diennakts laikā, jo tā ir telefona programma, kuras pamatfunkcija var tikt izsaukta jebkurā diennakts laikā, kad lietotājs to ir izvēlējies.

### 5.3. Lietojamība

Sistēma ir paredzēta lietotājiem, kuriem ir kaut kāda veida pieredze ar viedtālruni un skrārienjūtīgo ekrānu. Lietotāji, kuriem ir augstāk minētā pieredze, varēs strādāt ar sistēmu, bez papildus zināšanu iegūšanas.

### 5.4. Datu drošība

Programma izmanto datubāzi, kuru glabā ierīcē. Datubāzē glabājas modinātāji, to signāli ( tiek izmantoti jau ierīcē esošie signāli ), kā arī modinātājam pievienojamie uzdevumi. Lietotājam ir iespēja rediģēt tikai esošo modinātāju tabulu, taču pa vienam elementam - pievienot, labot vai dzēst vienu modinātāju, nav iespējams dzēst visu tabulu un zaudēt datus. Šāda veida sistēmai nav nepieciešams izstrādāt īpašu datu aizsardzību, taču ir padomāts par to, lai lietotājs tomēr nevarētu pazaudēt veselas datu tabulas uzreiz, un nevarētu mainīt uzdevumu tabulas, jo tas nav paredzēts.

### 5.5. Uzticamība

Sistēma izveidota tā, ka lietotājam nav daudz iespēju ievadīt kļūdainus datus. Izmantotie dati, ko lietotājs var ievadīt ir tekstuāli un tiek izmantoti attēlošanai, vai arī ir izvēle no saraksta, kas tiek izveidots sistēmā. Sistēma neizmanto nekāda veida lietotāja datus, kurus vajadzētu uzglabāt un par kuru drošību vajadzētu uztraukties.

# PROGRAMMATŪRAS PROJEKTĒJUMA APRAKSTS

## Ievads

Šajā nodaļā tiek aprakstīts dotais dokuments, tiek definēta mērķauditorija, tiek aprakstīta saistība ar citiem dokumentiem, aprakstīts dokumenta nolūks.

### 1.1. Nolūks

Šī dokumenta nolūks ir aprakstīt " Android lietotne – uzlabots modinātājs " sistēmas uzbūvi. Dokuments paredzēts sistēmas izstrādātājiem. Dokumenta nolūks ir palīdzēt sistēmas analīzē, plānošanā un implementēšanā.

### 1.2. Darbības sfēra

Šī lietojumprogramma ir paredzēta lietotājiem, kurus neapmierina iebūvētā modinātāja funkcionalitāte. Viņiem būs iespēja izvēlēties dažādus uzstādījumus, un pielāgot programmu savām vajadzībām, lai tā noderētu pēc iespējas labāk.

### 1.3. Definīcijas

#### 1.3.1. Izmatotie saīsinājumi

|  |  |
| --- | --- |
| **Saīsinājums** | **Skaidrojums** |
| PPS | Programmatūras prasību specifikācija |
| LVS | Latvijas valsts standarts |
| Java | Valoda, kas izmantota programmas funkcionalitātes nodrošināšanai. |
| XML | Valoda, kas, izmantojot speciālus kodus, nosaka ekrānformu izskatu lietotāja telefona ekrānā. |
| CRUD | Izveidošanas, lasīšanas, atjaunošanas un dzēšanas funkcionalitātes saīsinājums. (No angļu valodas – Create, Read, Update, Delete.) |

#### 1.3.2. Izmantotie jēdzieni

|  |  |
| --- | --- |
| **Jēdziens** | **Skaidrojums** |
| Lietotājs | Persona, kas izmanto lietojumprogrammu. |
| Serviss | Programmas sastāvdaļa, kas spēj darboties neatkarīgi no programmas, ja nu vienīgi to izmaina vai dzēš. Konkrētajā gadījumā šis palīdz modinātājam sākt darboties vajadzīgajā laika momentā. |

### 1.4. Saistība ar citiem dokumentiem

Šis dokuments tika izstrādāts pēc LVS 72:1996 "Ieteicamā prakse programmatūras projektējuma aprakstīšanai" standarta. Dokuments tika izveidots balstoties uz agrāk izstrādāto PPS. Dokumentu nepieciešams lietot kopā ar PPS.

## 2. Dekompozīcijas apraksts

### 2.1. Moduļu dekompozīcija

Moduļi ir sadalīti četru veidu moduļos:

**Kontrolieris** - klase, kas satur operāciju loģisko secību. Tajā atrodas pārsvarā tikai loģiskas darbības, un tā izmanto modeļu moduļus, lai veikt tehniskas darbības tādas, kā datu atlasīšana. Atgriež rezultātu lietotāja pieprasījumiem. Realizācijas valoda – Java.

**Modelis** - klase, kas satur metodes, kurās paredzētas datubāzes datu apstrādei. Realizācijas valoda – Java.

**Galvenais kontrolieris** – ir tikai viens tāds kontrolieris, atrodas tajā pašā mapē, kur pārējie (src/com/WMU/wakemeup/AlarmListActivity).

**Skats** – atbild par lietotāja ekrānformas izveidošanu, izmainīšanu vai aizpildīšanu. Glabājas mapē ( res/layout ). Realizācijas valoda XML

#### 2.1.1 Galvenais kontrolieris

|  |
| --- |
| Fails |
| AlarmListActivity.java |
| Apraksts |
| Uzsāk lietotāja un sistēmas sadarbību. Nodrošina sākuma skata funkcionalitāti, attēlo modinātājus, ļauj pievienot jaunu modinātāju, rediģēt eksistējošu modinātāju, dzēst eksistējošu modinātāju. |

#### 2.1.2. Pievienošanas labošanas kontrolieris

|  |
| --- |
| Fails |
| AlarmDetailsActivity.java |
| Apraksts |
| Nodrošina modinātāja pievienošanas/rediģēšanas funkcionalitāti. |

#### 2.1.3. Galvenā kontroliera palīgkontrolieris

|  |
| --- |
| Fails |
| AlarmListAdapter.java |
| Apraksts |
| Nodrošina modinātāju saraksta elementu pareizo izskatu, aktivizēšanas/deaktivizēšanas funkcionalitāti un modinātāja dzēšanas funkcionalitātes izsaukšanu. |

#### 2.1.4. Datubāzes un servisu apstrādājošais kontrolieris

|  |
| --- |
| Fails |
| AlarmManagerHelper.java |
| Apraksts |
| Nodrošina galvenā kontroliera komunikāciju ar datubāzi un modinātāja servisu. |

#### 2.1.5. Datu modelis

|  |
| --- |
| Fails |
| AlarmModel.java |
| Apraksts |
| Nodrošina datu pārnešanu, īslaicīgu uzglabāšanu. |

#### 2.1.6. Modinātāja izsaukšanas kontrolieris

|  |
| --- |
| Fails |
| AlarmScreen.java |
| Apraksts |
| Nodrošina modinātāja funkcijas izpausmes – „pamodina” ierīci, atskaņo modinātāja signļālu veic nepieciešamo funkcionalitāti modināšanas procesam. |

#### 2.1.7. Modinātāja servisa izveidošanas kontrolieris

|  |
| --- |
| Fails |
| AlarmService.java |
| Apraksts |
| Nodrošina modinātāja darbības uzsākšanu uzstādītajā laika momentā. |

#### 2.1.8. Slēdža pielāgošanas kontrolieris

|  |
| --- |
| Fails |
| CustomSwitch.java |
| Apraksts |
| Pārveido izvēles rūtiņu par „Ieslēgts/Izslēgts” slēdzi, nodrošina tā funkcionalitāti. |

#### 2.1.9. Datubāzes kontrolieris

|  |
| --- |
| Fails |
| AlarmDBHelper.java |
| Apraksts |
| Nodrošina datubāzes realizāciju un CRUD funkcionalitāti. |

### 2.2. Datu dekompozīcija

Datu bāzē tiks veidotas šādas datu tabulas:

|  |  |
| --- | --- |
| **Datu tabula** | **Tabulas nolūks** |
| alarms | Tabulas nolūks ir uzglabāt informāciju par lietotāja izveidotajiem modinātājiem. Katram modinātājam tiks veidota sava tabula. |
| counting\_tasks\_arabicnr | Tabulas nolūks ir uzglabāt saskaitīšanas uzdevumus un to atbildes. Tie ir attēloti ar arābu cipariem. |
| multiply\_tasks\_arabicnr | Tabulas nolūks ir uzglabāt reizināšanas uzdevumus un to atbildes. Tie ir attēloti ar arābu cipariem. |
| substraction\_tasks\_arabicnr | Tabulas nolūks ir uzglabāt atņemšanas uzdevumus un to atbildes. Tie ir attēloti ar arābu cipariem. |
| dividing\_tasks\_arabicnr | Tabulas nolūks ir uzglabāt dalīšanas uzdevumus un to atbildes. Tie ir attēloti ar arābu cipariem. |
| counting\_tasks\_romannr | Tabulas nolūks ir uzglabāt saskaitīšanas uzdevumus un to atbildes. Tie ir attēloti ar romiešu cipariem. |
| multiply\_tasks\_romannr | Tabulas nolūks ir uzglabāt reizināšanas uzdevumus un to atbildes. Tie ir attēloti ar romiešu cipariem. |
| substraction\_tasks\_romannr | Tabulas nolūks ir uzglabāt atņemšanas uzdevumus un to atbildes. Tie ir attēloti ar romiešu cipariem. |
| dividing\_tasks\_romannr | Tabulas nolūks ir uzglabāt dalīšanas uzdevumus un to atbildes. Tie ir attēloti ar romiešu cipariem. |

## 3. Atkarības apraksts

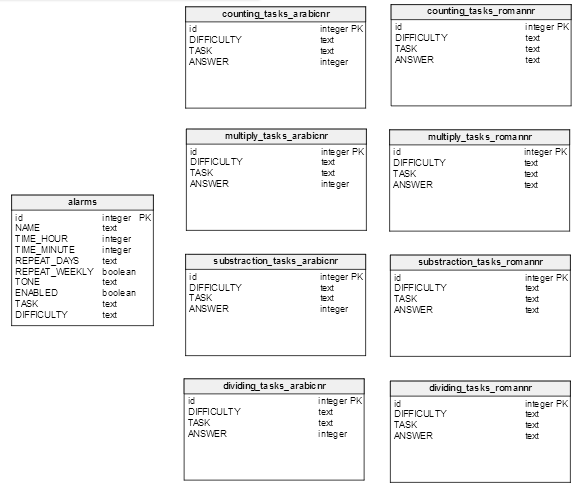
### 3.1. Starpmoduļu atkarības

Tabulā tiks izmantoti moduļu failu nosaukumi.

|  |  |  |
| --- | --- | --- |
| **Modulis** | **Izmanto moduļus** | **Tiek izmantots moduļos** |
| AlarmContract.java | - | AlarmDBHelper.java |
| AlarmDBHelper.java | AlarmModel.java | AlarmDetailsActivity.java, AlarmListActivity.java, AlarmManagerHelper.java |
| AlarmDetailsActivity.java | AlarmDBHelper.java, AlarmModel.java, AlarmManagerHelper.java | AlarmListActivity.java, AlarmListAdapter.java |
| AlarmListActivity.java | AlarmListAdapter.java, AlarmDBHelper.java, CustomSwitch.java | AlarmListAdapter.java |
| AlarmListAdapter.java | AlarmModel.java, AlarmListActivity.java | AlarmListActivity.java |
| AlarmManagerHelper.java | AlarmModel.java, AlarmService.java, AlarmContract.java | AlarmDetailsActivity.java, AlarmService.java, AlarmScreen.java |
| AlarmModel.java | - | AlarmDetailsActivity.java  AlarmListActivity.java  AlarmDBHelper.java, AlarmListAdapter.java, AlarmManagerHelper.java |
| AlarmScreen.java | AlarmManagerHelper.java | AlarmService.java |
| AlarmService.java | AlarmScreen.java, AlarmManagerHelper.java | AlarmManagerHelper.java |
| CustomSwitch.java | - | AlarmDetailsActivity.java |

### 3.2. Datu atkarības

Šajā sadaļā tiek attēlotas datubāzes tabulu savstarpējās atkarībass. Attēlā 3.2. redzams, ka tabulas ir neatkarīgas viena no otras.



3.2.att

## 4. Detalizēts projektējums

### 4.1. Datu detalizēts projektējums

### 4.1.1. Tabula „alarms”

Tabulā Alarms tiek glabāta informācija par konkrētu modinātāju.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lauka nosaukums** | **Datu tips** | **Obligāts** | **Noklusētā vērtība** | **PK/FK/UK** | **Apraksts** |
| ID | Integer | Jā | - | PK | Unikāls identifikators. Tiek ģenerēts automātiski. |
| NAME | Text | Nē | - |  | Modinātāja nosaukums. |
| TIME\_HOUR | Integer | Jā | - |  | Stunda |
| TIME\_MINUTE | Integer | Jā | - |  | Minūte |
| REPEAT\_DAYS | Text | Jā | - |  | Atkārtošanas dienas |
| REPEAT\_WEEKLY | Boolean | Jā | - |  | Nedēļas atkārtojums |
| TONE | Text | Jā | - |  | Skaņas signāls |
| ENABLED | Boolean | Jā | - |  | Aktīvs vai neaktīvs |
| TASK | TEXT | Nē | - |  | Uzdevuma tips |
| DIFFICULTY | TEXT | Nē | - |  | Uzdevuma sarežģītība |

### 4.1.2. Tabula „counting\_tasks\_arabicnr”

Tabulā „counting\_tasks\_arabicnr” tiek glabāti saskaitīšanas uzdevumi ar arābu cipariem.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lauka nosaukums** | **Datu tips** | **Obligāts** | **Noklusētā vērtība** | **PK/FK/UK** | **Apraksts** |
| ID | Integer | Jā | - | PK | Unikāls identifikators. Tiek ģenerēts automātiski. |
| DIFFICULTY | Text | Jā | - |  | Uzdevuma sarežģītības pakāpe |
| TASK | Text | Jā | - |  | Uzdevuma saturs |
| ANSWER | Integer | Jā | - |  | Uzdevuma atbilde |

### 4.1.3. Tabula „multiply\_tasks\_arabicnr”

Tabulā „counting\_tasks\_arabicnr” tiek glabāti reizināšanas uzdevumi ar arābu cipariem.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Lauka nosaukums | Datu tips | Obligāts | Noklusētā vērtība | PK/FK/UK | Apraksts |
| ID | Integer | Jā | - | PK | Unikāls identifikators. Tiek ģenerēts automātiski. |
| DIFFICULTY | Text | Jā | - |  | Uzdevuma sarežģītības pakāpe |
| TASK | Text | Jā | - |  | Uzdevuma saturs |
| ANSWER | Integer | Jā | - |  | Uzdevuma atbilde |

### 4.1.4. Tabula substraction\_tasks\_arabicnr

Tabulā „counting\_tasks\_arabicnr” tiek glabāti atņemšanas uzdevumi ar arābu cipariem.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lauka nosaukums** | **Datu tips** | **Obligāts** | **Noklusētā vērtība** | **PK/FK/UK** | **Apraksts** |
| ID | Integer | Jā | - | PK | Unikāls identifikators. Tiek ģenerēts automātiski. |
| DIFFICULTY | Text | Jā | - |  | Uzdevuma sarežģītības pakāpe |
| TASK | Text | Jā | - |  | Uzdevuma saturs |
| ANSWER | Integer | Jā | - |  | Uzdevuma atbilde |

### 4.1.5. Tabula dividing\_tasks\_arabicnr

Tabulā „counting\_tasks\_arabicnr” tiek glabāti dalīšanas uzdevumi ar arābu cipariem.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lauka nosaukums** | **Datu tips** | **Obligāts** | **Noklusētā vērtība** | **PK/FK/UK** | **Apraksts** |
| ID | Integer | Jā | - | PK | Unikāls identifikators. Tiek ģenerēts automātiski. |
| DIFFICULTY | Text | Jā | - |  | Uzdevuma sarežģītības pakāpe |
| TASK | Text | Jā | - |  | Uzdevuma saturs |
| ANSWER | Integer | Jā | - |  | Uzdevuma atbilde |

### 4.1.6. Tabula counting\_tasks\_romannr

Tabulā „counting\_tasks\_arabicnr” tiek glabāti saskaitīšanas uzdevumi ar romiešu cipariem.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lauka nosaukums** | **Datu tips** | **Obligāts** | **Noklusētā vērtība** | **PK/FK/UK** | **Apraksts** |
| ID | Integer | Jā | - | PK | Unikāls identifikators. Tiek ģenerēts automātiski. |
| DIFFICULTY | Text | Jā | - |  | Uzdevuma sarežģītības pakāpe |
| TASK | Text | Jā | - |  | Uzdevuma saturs |
| ANSWER | Text | Jā | - |  | Uzdevuma atbilde |

### 4.1.7. Tabula multiply\_tasks\_romannr

Tabulā „counting\_tasks\_arabicnr” tiek glabāti reizināšanas uzdevumi ar romiešu cipariem.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lauka nosaukums** | **Datu tips** | **Obligāts** | **Noklusētā vērtība** | **PK/FK/UK** | **Apraksts** |
| ID | Integer | Jā | - | PK | Unikāls identifikators. Tiek ģenerēts automātiski. |
| DIFFICULTY | Text | Jā | - |  | Uzdevuma sarežģītības pakāpe |
| TASK | Text | Jā | - |  | Uzdevuma saturs |
| ANSWER | Text | Jā | - |  | Uzdevuma atbilde |

### 4.1.8. Tabula substraction\_tasks\_romannr

Tabulā „counting\_tasks\_arabicnr” tiek glabāti atņemšanas uzdevumi ar romiešu cipariem.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lauka nosaukums** | **Datu tips** | **Obligāts** | **Noklusētā vērtība** | **PK/FK/UK** | **Apraksts** |
| ID | Integer | Jā | - | PK | Unikāls identifikators. Tiek ģenerēts automātiski. |
| DIFFICULTY | Text | Jā | - |  | Uzdevuma sarežģītības pakāpe |
| TASK | Text | Jā | - |  | Uzdevuma saturs |
| ANSWER | Text | Jā | - |  | Uzdevuma atbilde |

### 4.1.9. Tabula dividing\_tasks\_romannr

Tabulā „counting\_tasks\_arabicnr” tiek glabāti dalīšanas uzdevumi ar romiešu cipariem.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lauka nosaukums** | **Datu tips** | **Obligāts** | **Noklusētā vērtība** | **PK/FK/UK** | **Apraksts** |
| ID | Integer | Jā | - | PK | Unikāls identifikators. Tiek ģenerēts automātiski. |
| DIFFICULTY | Text | Jā | - |  | Uzdevuma sarežģītības pakāpe |
| TASK | Text | Jā | - |  | Uzdevuma saturs |
| ANSWER | Text | Jā | - |  | Uzdevuma atbilde |

# TESTĒŠANAS DOKUMENTĀCIJA

## Ievads

Šajā dokumentā ir aprakstīti sistēmas „Androd lietojumprogramma – uzlabots modinātājs” testēšanas rezultāti. Testēšana notika gan pēc katras funkcijas realizēšanas, gan kad tika izstrādāta visa sistēma, lai pārbaudītu vai viss strādā atbilstoši sagaidītajam rezultātam.

## Testēšanas rezultāti

Testēšanas rezultāti tika apkopoti tabulā ar šādiem laukiem:

* Nr. – numurs, kas identificē konkrēto testu;
* Testa apraksts – apraksta testa darbības konkrētajam testējamajam vienumam;
* Sagaidāmais rezultāts – apraksts par to, kas tiek sagaidīts. Ja sagaidāmais rezultāts ietver sevī kādu paziņojumu, tad tas tiek norādīts kā identifikators ( apskatīt paziņojumus var pielikuma sadaļā „Paziņojumi” );
* Statuss – norāda uz to vai testa rezultāts atbilst sagaidītajam, ja jā, tad laukā ieraksta +, ja nē, apraksta problēmu un vai problēma ir atrisināta;

Testa apraksta norādes (X) norāda uz to, kādu testu jāizpilda pirms šī testa, un ka šis tests ir turpinājums testa X veiktai darbībai(testa X testa apraksts un dotā testa apraksts). Pēc katra vienuma notestēšanas tiek dots problēmu kopsavilkums, kurā apraksta radušas problēmas testpiemēros, un vai tas tika novērstas.

Testēšanā vienumi tiks pārbaudīti uz parastiem datiem, robeţvērtībām, tukšiem laukiem, nekorektiem datiem. Tiks pārbaudīta arī lietotāja saskarne – izskats un funkcionalitāte.

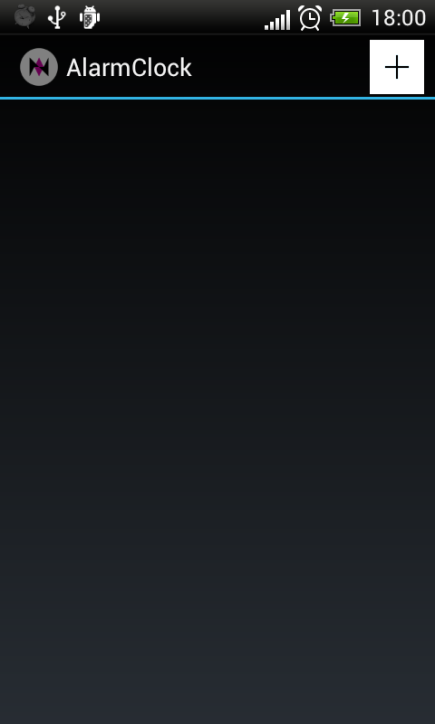
|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Testa apraksts** | **Sagaidāmais rezultāts** | **Statuss** |
| 1 | Telefonam esot gaidīšanas stāvoklī, atver programmu. | Programma atveas. Tiek atvērts galvenais skats. ( vajag pielikumu ) Ja ir saglabāts kāds vai vairāki modinātāji – tiek attēlots modinātāju saraksts. | + |
| 2 | Telefona „Darbību joslā” tiek nospiesta poga „+” | Tiek atvērts skats, kurā var rediģēt rediģēt informāciju par modinātāju un to pēc tam saglabāt. | + |
| 3 | Izvēlēties modinātāja darbības dienas nedēļā. Pieskaršanās laukam, kur ir dienu uzskaitījums. | Tiek atvērts skats, kurā var izvēlēties, kurās dienās konkrētais modinātājs veiks modināšanas darbību. | + |
| 4 | Izvēlēties modinātāja skaņas signālu. Pieskaršanās laukam, kur ir modinātātāja skaņas signāla nosaukums. | Tiek atvērts skats, kurā var izvēlēties kāds būs konkrētā modinātāja signāls. | + |
| 5 | Ekrāna satura ritināšana. ( ja ekrāna saturs neietilpst ekrāna augstumā ) | Pieskaroties ekrānam un velkot ar pirkstu uz augšu, vai uz leju ekrānā attēlotais saturs tiek mainīts – tas, kas līdz šim nebija redzams kļūst redzams. | Sākotnēji neritinājās, salabots. |
| 6 | Modinātāja deaktivizēšana | Pieskaroties pogai, kas ir blakus konkrētajam modinātājam sarakstā, tā no maina stāvokli no zaļa uz pelēku un attiecīgais modinātājs arī vairs nav aktīvs – nemodina saglabātajās dienās konkrētajā laikā. | + |
| 7 | Modinātāja aktivizēšana | Pieskaroties pogai, kas ir blakus konkrētajam modinātājam sarakstā, tā no maina stāvokli no pelēka uz zaļu un attiecīgais modinātājs kļūst aktīvs – modina saglabātajās dienās konkrētajā laikā. | + |
| 8 | Rediģēt modinātāju | Izvēloties kādu modinātāju no saraksta ( pieskaroties tam ) tiek atvērts modinātāja pievienošanas/rediģēšanas skats ar konkrētajam modinātājam atbilstošo informāciju. Mainot informāciju un nospiežot pogu saglabāt tā tiek saglabāta un modinātājs ir aktīvs ( arī tad, ja iepriekš bija deaktivizēts ) | + |
| 9 | Dzēst modinātāju | Izpildot ilgāku pieskārienu, parādās skats, kurā lietotājam jautā, vai viņš vēlas dzēst konkrēto modinātāju. | + |
| 10 | Apstiprināt modinātāja dzēšanu | 9.testpiemērā minētajā skatā izvēloties dzēst, modinātājs tiek izdzēsts, tiek atvērts galvenais skats, kurā vairs nav šis modinātājs. | + |
| 11 | Atcelt modinātāja dzēšanu | 9.testpiemērā minētajā skatā izvēloties atcelt, tiek atvērts galvenais skats, kurā joprojām ir šis modinātājs un citi ( ja tādi iepriekš bija ). | + |
| 12 | Atcelt modinātāja pievienošanu | Modinātāja pievienošanas skatā nospiežot pogu atpakaļ lietotājs atgriežas sākuma skatā, nesaglabājot izmaiņas. | + |
| 13 | Atcelt modinātāja rediģēšanu | Modinātāja pievienošanas skatā nospiežot pogu atpakaļ lietotājs atgriežas sākuma skatā, nesaglabājot izmaiņas. | + |
| 14 | Izvēlēties atkārtošanas dienas | Pieskaroties laukam, kurā ir uzskaitītas atkārtošanas dienas (pievienošanas/ rediģēšanas skatā), tiek atvērts jauns skats, kurā var izvēlēties atkārtošanas dienas atzīmējot izvēles rūtiņas. | + |
| 15 | Rediģēt atkārtošanas dienas | Pieskaroties laukam, kurā ir uzskaitītas atkārtošanas dienas (pievienošanas/ rediģēšanas skatā), tiek atvērts jauns skats, kurā var izvēlēties atkārtošanas dienas atzīmējot izvēles rūtiņas. Iepriekš atzīmētās dienas jau ir atzīmētas. | + |
| 16 | Izvēlēties modinātāja signālu | Pieskaroties laukam, kurā ir redzams modinātāja signāls (pievienošanas/ rediģēšanas skatā), tiek atvērts jauns skats, kurā var izvēlēties modinātāja signālu no saraksta atzīmējot to ar „radio pogu.” | + |
| 17 | Rediģēt modinātāja signāla izvēli | Pieskaroties laukam, kurā ir redzams modinātāja signāls (pievienošanas/ rediģēšanas skatā), tiek atvērts jauns skats, kurā var izvēlēties modinātāja signālu no saraksta atzīmējot to ar „radio pogu.” Iepriekš izvēlētais signāls ir atzīmēts atverot šo skatu. | + |
| 18 | Izvēlēties izpildāmā uzdevuma tipu | Pieskaroties laukam, kurā ir redzams modinātāja uzdevuma tips (pievienošanas/ rediģēšanas skatā), tiek atvērts jauns skats, kurā var izvēlēties modinātāja uzdevuma tipu no saraksta atzīmējot to ar „radio pogu.” | + |
| 19 | Rediģēt izpildāmā uzdevuma tipa izvēli | Pieskaroties laukam, kurā ir redzams modinātāja uzdevuma tips (pievienošanas/ rediģēšanas skatā), tiek atvērts jauns skats, kurā var izvēlēties modinātāja uzdevuma tipu no saraksta atzīmējot to ar „radio pogu.” Iepriekš izvēlētais tips ir atzīmēts atverot šo skatu. | + |
| 20 | Izvēlēties izpildāmā uzdevuma sarežģītību | Pieskaroties laukam, kurā ir redzams modinātāja uzdevuma sarežģītības pakāpe (pievienošanas/ rediģēšanas skatā), tiek atvērts jauns skats, kurā var izvēlēties modinātāja uzdevuma sarežģītības pakāpi no saraksta atzīmējot to ar „radio pogu.” |  |
| 21 | Rediģēt izpildāmā uzdevuma sarežģītības izvēli | Pieskaroties laukam, kurā ir redzams modinātāja uzdevuma sarežģītības pakāpe (pievienošanas/ rediģēšanas skatā), tiek atvērts jauns skats, kurā var izvēlēties modinātāja uzdevuma sarežģītības pakāpi no saraksta atzīmējot to ar „radio pogu.” Iepriekš izvēlētā sarežģītības pakāpe ir atzīmēta atverot šo skatu. | + |
| 22 | Izslēgt modinātāju, kad tas ir sācis darboties ( bez uzdevuma ) | Sākoties modinātājam ir iespēja izslēgt modinātāju ( poga Dismiss ), pieskaroties šai pogai modinātājs tiek izslēgts. | + |

Novērojumi - testējot ar dažādām ierīcēm dažas funkcionalitātes daļas minimāli atšķīrās. Laika izvēle ar HTC ONE V un Just5 Spacer notika atšķirīgi. HTC ONE V bija problēmas ar ritošo laika izvēli, un tā vietā es izvēlējos lietot bultiņas virs un zem ritošās joslas, taču Just5 nebija ne mazāko problēmu ar ritošajām daļām.

# EKRĀNFORMU PIEMĒRI

## Sākuma skats

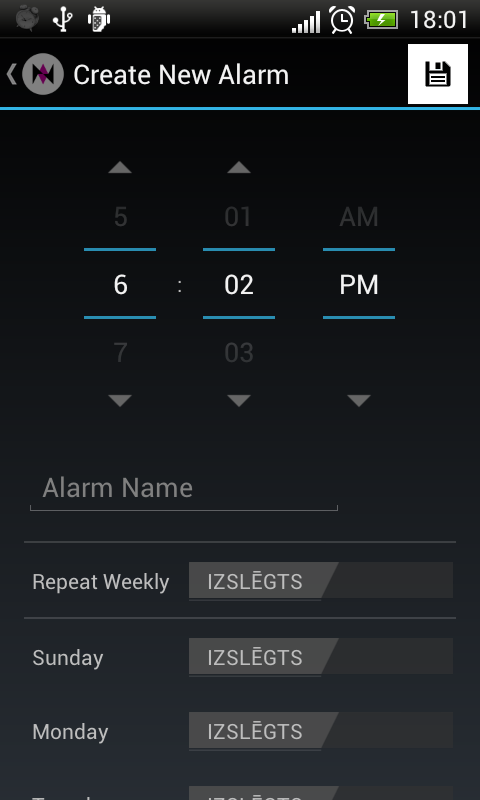
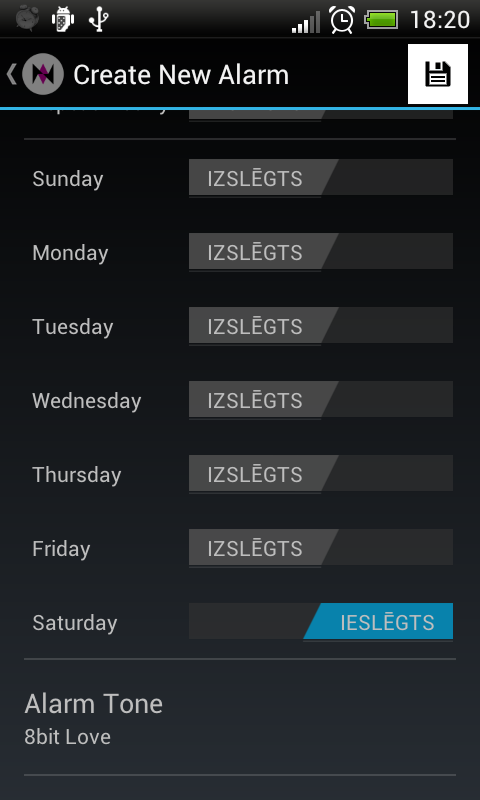
Kad sistēma ir ielādēta, lietotājam tiek parādīta sākuma skats, kurā tiktu attēloti modinātāji. Tā kā tikko izveidotā programā modinātāji vēl nav, tāpēc šobrīd ir tukšs laukums. Augšējā daļā ir redzama darbību josla ar programmas ikonu un nosaukumu kreisajā pusē un labajā poga „+”, ar kuru var pievienot jaunu modinātāju.



1.att. Sākuma skats.

## Modinātāja pievienošanas/rediģēšanas skats

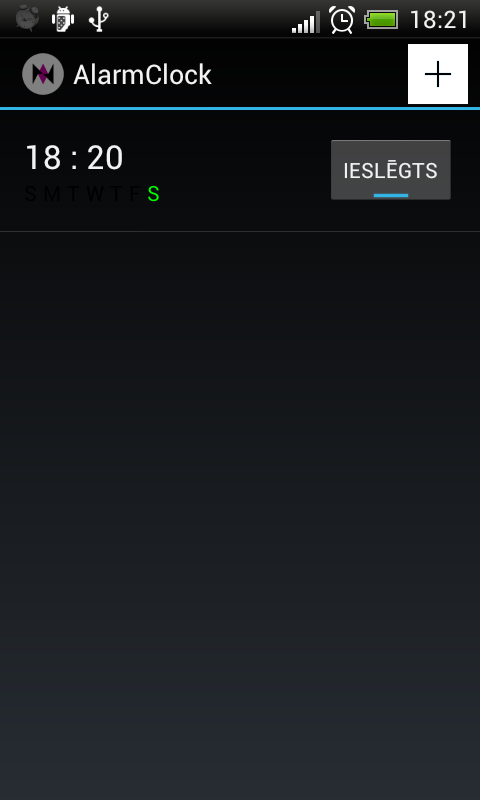
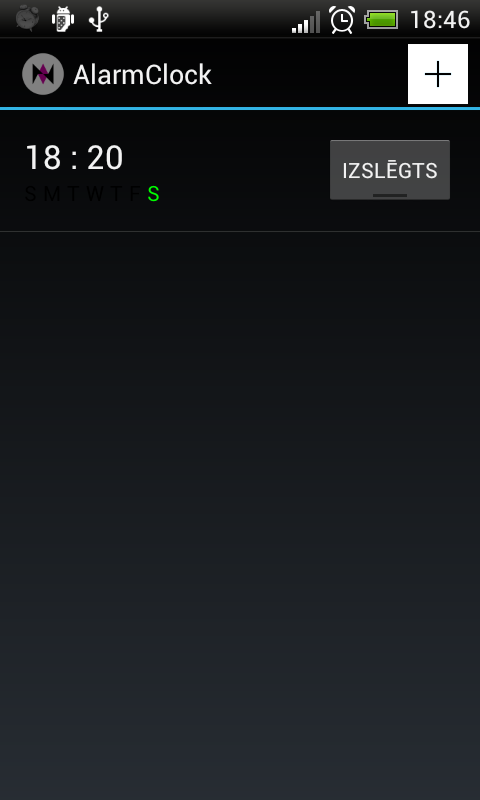
Pēc „+” nospiešanas tiek atvērts pievienošanas/rediģēšanas skats. Skata augšējā daļā ir redzama darbību josla, kuras kreisajā iespēja iet atpakaļ uz sākuma skatu „<” un labajā pusē ir iespēja saglabāt modinātāju. Zem darbību joslas redzami visi iespējamie uzstādījumi, kurus var mainīt konkrētajam modinātājam. Pēc modinātāja saglabāšanas tiek atvērts sākuma skats ar modinātāju sarakstu, tikko izveidotais modinātājs jau ir pievienots sarakstam.

2.att. Modinātāja pievienošanas/rediģēšanas skats.

## Modinātājs saglabāts

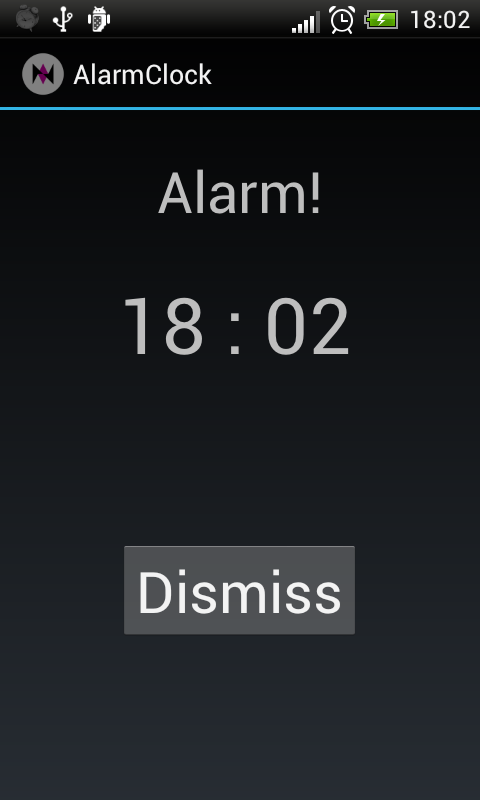
Kad modinātājs ir saglabāts, tas uzrādās sākuma skatā tā, kā tas ir attēlots attēlā pa kreisi. Ja lietotājs vēlas, viņš var modinātāju deaktivizēt, mainīsies pogas stāvoklis no „Ieslēgts” uz „Izslēgts” kā tas ir redzams attēlā pa labi.

3.att. Modinātājs saglabāts, aktīvs ( pa kreisi ), neaktīvs ( pa labi ).

## Modinātāja aktivizēšanās skats – modināšana.

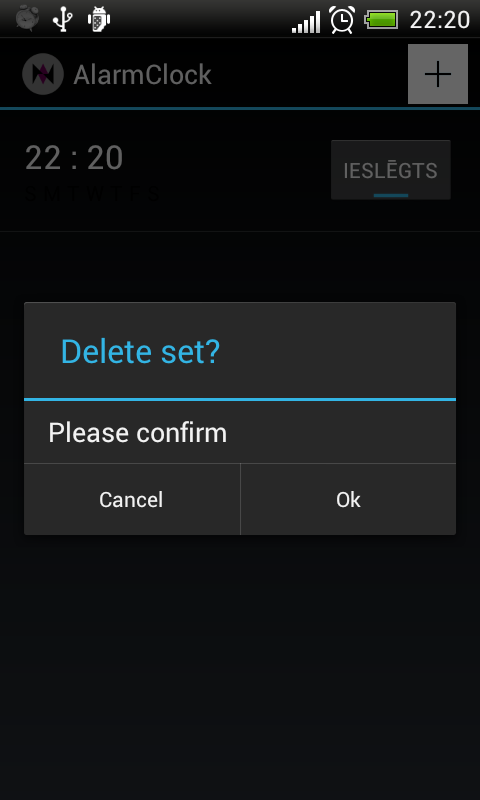
Šajā skatā redzams kā izskatās modinātājs modināšanas skatā ( bez uzdevuma ). Pēc pogas „Dismiss” nospiešanas, modinātājs tiek izslēgts un tiek rādīts visu modinātāju saraksts pat, ja programma pirms tam nebija atvērta.



4.att. Modinātāja aktivizēšanās skats – modināšana.

## Modinātāja dzēšana

Sākuma skatā izpildot ilgo pieskārienu kādam modinātājam, tiek atvērts skats un programma jautā vai izdzēst konkrēto modinātāju. Spiešot pogu „Ok”, tiek dzēsts modinātājs, spiežot „Cancel” tas netiek dzēsts. Pēc jebkura no gadījumiem šis skats aizveras un tiek rādīts sākuma skats ar modinātāju sarakstu.



5.att. Modinātāja dzēšana

# PROJEKTA ORGANIZĀCIJA

Projekta organizācija tika izpildīta pēc ūdenskrituma dzīves cikla modeļa. Tika izvirzītas pirmās idejas par to, ko produkts varētu darīt. Pēc ideju apkopošanas, tika izvēlētas svarīgākās, un tādas, kuras var realizēt atvēlētajā laikā. No šīm idejām bija izvirzītas prasības, kuras tika apkopotas „Programmatūras prasību specifikācijā”. Pēc prasību apkopošanas, tika uzrakstīts sistēmas projektējums, pēc kura tika būvēta sistēma. Izstrāde bija uzsākta ar moduļu izstrādi, kas atbild par modinātāja pamatfunkciju nodrošināšanu sistēmā. Pēc funkciju realizēšanas tika pārbaudīta sistēmas darbība, kuras rezultātā varēja redzēt sistēmas uzvedību uz mobilā tālruņa. Tika izstrādāts jauns(uzlabots) „Programmatūras projektējuma apraksts”, veiktas izmaiņas sistēmas uzbūvē, lai uzlabotu sistēmas darbību un vizuālo noformējumu.

Izstrādātajai sistēmai pasūtītāja nav, tā ir paša izdomāta lietojumprogrammatūra, tāpēc pats izvirzīju prasības sistēmai, kuras apkopoju programmatūras prasību specifikācijā un pēc šī dokumenta izstrādāju arī programmatūras projektējuma aprakstu. Pēc šiem dokumentiem tika uzprogrammēta sistēma un notestēta.

Šo projektu izstrādāju patstāvīgi, bet idejas un ieteikumus sistēmas izstrādei meklēju internetā kā arī saņēmu no kvalifikācijas darba vadītāja.

# KVALITĀTES NODROŠINĀŠANA

Lai nodrošinātu izstrādātā projekta kvalitāti, tika veiktas šādas darbības:

1. Programmatūras dokumentācija tika veikta saskaņā ar valsts standartiem:

* LVS 68:1996 Programmatūras prasību specifikācijas ceļvedis,
* LVS 72:1996 Ieteicamā prakse programmatūras projektējuma aprakstīšanai.

2. Izstrādātājs kods tika būvēts noteiktā struktūrā, lai līdzīgo moduļu starpā, tas būts strukturēts un noformēts vienādi, lai uzlabotu koda saprotamību.

3. Koda daļas, kas ir līdzīgas un atkārtojas vairākos moduļos, tika iznestās atsevišķās funkcijās, kas atviegloja sistēmas būvēšanu un modificēšanu.

4. Sistēma tika būvēta tā, lai to varētu viegli modificēt, pielikt jaunus moduļus klāt, vai uzlabot jau esošos.

5. Lietotāja saskarne veidotā vienotā, nepārprotamā stilā, intuitīvi viegli uztverama. Saskarne ir būvēta tā, lai būtu viegli strādāt ar sistēmu tieši uz mobilā tālruņa.

6. Notika dažādu lietošanas scenāriju testēšana.

# KONFIGURĀCIJU PARVALDĪBA

Tā kā programmprodukta izstrādē piedalījās tikai viens programmētājs, speciāli konfigurācijas pārvaldības rīki netika izmantoti, bet tika izmantots tikai versiju vadības rīks Github. Izmantojot šo rīku, varēja iegūt sistēmas izmaiņu vēsturi, un sistēmas koda sabojāšanas rezultāta, atgriezties pie kādas no iepriekšējam versijām. Sistēmas izstrāde notika uz izstrādātāja datora. Kad bija izveidots kāds no moduļiem, lai pārbaudīt tā darbību uz mobilā tālruņa, sistēmas izmaiņas tika saglabātas uz servera repozitorijā izmantojot Github. Papildus pašas sistēmas koda saglabāšanai uz servera, bija divas datubāzes, viena uz izstrādātāja datora, otra uz servera, ka arī uz papildus datu nesējiem tika glabātas noeksportētais datubāzes saturs.

Sistēmas versijas numurs veidots no trim cipariem x.y.z , kur x norāda uz lietojumprogrammatūras paaudzi, y – lielām izmaiņām sistēmas uzbūvē un izmaiņām tās prasībās pret to, z - nelielām izmaiņām programmas kodā.

Pašreizējā sistēmas versija ir 1.1.0.

# DARBIETILPĪBAS NOVĒRTĒJUMS

Darbietilpības novērtēta aptuveni izmantojot COCOMO 81 Intermediate Model Implementation.[5.]

Metodes formula: E=ai(LOC/1000)^(bi)\*EAF, kur

E – Programmas izstrādei nepieciešamie personmēneši

LOC = nepieskaņoto funkcijpunktu skaits \* vidējs programmrindiņu skaits viena funkcijpunkta realizācijai.

EAF (effort adjustment factor) – aprēķināts pēc formulas EAF = E1\*E2\*…\*E15.

ai, bi – koeficienti, kas ir atkarīgi no projekta tipa.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Sarežģīti** | **Vidēji** | **Vienkārši** | **Kopā** |
| **Ievadi** | 6\*0 | 4\*0 | 3\*12 | 36 |
| **Izvadi** | 7\*1 | 5\*0 | 4\*13 | 52 |
| **Iekšējie** **datu** **faili** | 15\*0 | 10\*0 | 7\*9 | 63 |
| **Ārējās** **saskarnes** | 10\*0 | 7\*0 | 5\*0 | 0 |
| **Vaicājumi** | 6\*0 | 4\*1 | 3\*4 | 16 |
| **Nepieskaņotu funkcijpunktu skaits:** | | | | 167 |

Vidējais rindiņu skaits funkcijai Java: 53 [9.]

Tātad prognozējamais rindiņu skaits kodam (LOC) būs 53\*167=8851

|  |  |  |
| --- | --- | --- |
| **Parametrs** | **Vērtība** | |
| **Produkta atribūti** | | |
| Drošums | Vidējs | 1 |
| Datubāzes izmērs | Vidējs | 1 |
| Produkta sarežģītība | Zema | 0,85 |
| Datoru atribūti | | |
| Izpildes laika ierobežojumi | Vidēji | 1 |
| Atmiņas ierobežojumi | Augsti | 1,06 |
| Vides mainīgums | Zems | 0,87 |
| Telefona atbildes laiks | Augsts | 1,07 |
| **Personāla atribūti** | | |
| Analītiskās spējas | Vidējas | 1 |
| Pieredze lietotņu izstrādē | Zema | 1,13 |
| Programmētāju spējas | Vidējas | 1 |
| Izstrādes vides pieredze | Zema | 1,10 |
| Programmēšanas valodas pieredze | Zema | 1,07 |
| **Projekta atribūti** | | |
| Programmatūras izstrādes rīku izmantošana | Vidēja | 1 |
| Izstrādes rīku izmantošana | Vidēja | 1 |
| Izstrādes grafikas uzliktie ierobežojumi | Vidēja | 1 |
| **Reizinājums(EAF):** | | 1,1155 |

Projekta tips ir „Organic”, jo izstrādāju projektu viens pats. No tā izriet, ka ai = 3,2, bet bi=1,05

Risinājums: E = 3,2\*( 8851 / 1000 ) ^1,05\*1,1155 = 3,2\*8,851^1,1712= 41,14 personmēneši.

Papildus šim aprēķinam veicu arī savu aptuveno aprēķinu balstoties uz savām spējām un pieredzi.

Pirmkārt, iepriekš aprēķinātais rindiņu skaits kodā ir stipri lielāks par to, ko spētu izstrādāt trijos personmēnešos. Rindiņu skaita aprēķināšanai ņemšu moduļu skaitu sistēmā – 10 moduļi. Plānojot PPA biju sadalījis moduļus tā, ka katrā no viņiem rindiņu skaits vidēji varētu būt ap 100, jāņem vērā arī tas, ka bija nepieciešams izveidot arī skatu noformējumu izmanotjot XML, ieplānoju izveidot aptuveni 5 skatus, katrā vidēji ap 60 rindiņām. Sanāk, ka aptuveni paredzētais koda rindiņu skaits ir 1300. Ja ieliek šo skaitli iepriekš iegūtajā COCOMO formulā iegūst – E = 4,35 personmēneši.

# IZMANTOTĀS TEHNOLOĢIJAS

Sistēmas izstrādē tika izmantotas šādas tehnoloģijas:

* Izstrādes vide Eclipse
* Versijkontroles rīks GitHub
* Java
* XML
* SQLi

# SECINĀJUMI

Kvalifikācijas darbā izstrādāju mobilo tālruņu lietojumprogrammatūru, kuras galvenā funkcija ir pamodināt tālruņa īpašnieku tā, kā viņš to iepriekš ir uzstādījis, pat ja tie ir vairāki modinātāji pēc kārtas ar citādākiem uzdevumiem katru reizi.

Projekta izstrādes laikā bija iespēja iegūt jaunas zināšanas pašmācības ceļā, kādas nebija mācītas universitātē un nosacīti pielietot jau esošās. Šī bija pirmā reize Android lietojumprogrammas izstrādē. Paplašināju, ieguvu un atkārtoju zināšanas Java, XML un SQL tehnoloģiju izmantošanā. Ieguvu zināšanas par to kā notiek speciālu servisu izveidošana un saglabāšana. Iemācījos testēt programmu gan izmantojot virtuālo mašīnu, gan reālu ierīci, kas izrādījās daudz ērtāk un ātrāk, jo izmantoja ierīces resursus nevis datora.

Izstrādājot šo projektu vajadzēja atcerēties, ka sistēma ir domāta viedajiem tālruņiem, kuru ātrdarbība un atmiņas resursi ir stipri zemāki, nekā stacionāram vai portatīvajam datoram, tāpēc daļa laika tika pavadīta pie koda optimizēšanas, lai izmantotu pēc iespējas mazāku resursu apjomu, kā arī uzlabotu koda lasāmību. Projekta izstrādes laikā ieguvu pieredzi arī ar versijkontroles rīku github, kas ļoti noderēja, jo tieši notika nopietna probēma ar datoru un nevarēja iztikt bez pārinstalēšanas, iepriekšējie dati tika pazaudēti, taču visu varēju atgūt pēc darba vides uzstādīšanas un datora sagatavošanas darbam.

Šī projekta izstrādes rezultātā guvu plašākas zināšanas par projekta izstrādes procesu un programmēšanas iemaņas uzlabojās salīdzinoši ar tām, kas bija pirms šī projekta. Projekta izstrāde neapstāsies, jo šo programmu vēlos pilnveidot un vēlāk pievienot google play un piedāvāt iespēju citiem cilvēkiem to izmantot sākumā par brīvu un vēlāk pievienojot papildus funkcionalitāti dot iespēju iegādāties papildinājumus.

# IZMANTOTĀ LITERATŪRA

* <http://developer.android.com/reference/android/provider/AlarmClock.html>
* <http://stackoverflow.com/questions/9983806/alarmclock-for-beginners-android>
* <http://www.itcsolutions.eu/2011/08/31/android-tutorial-how-to-create-and-display-a-new-form-window-or-activity/>
* <http://stackoverflow.com/questions/8847171/android-timepicker-wheel-style-not-responding-correctly-to-flick-gestures-insi>
* <http://en.wikipedia.org/wiki/COCOMO>
* <http://www.steventrigg.com/data-persistence-using-sqlite-create-an-alarm-clock-in-android-tutorial-part-4/>
* <http://elietas.wordpress.com/2009/06/14/rekomendacijas-programmetaju-kvalifikacijas-darbiem/>
* <http://martinskemme.wordpress.com/2012/06/04/programmetaju-kvalifikacijas-darbi-2012/>
* <http://www.qsm.com/resources/function-point-languages-table>
* http://developer.android.com/reference/android/app/Service.html

## PIELIKUMI

## PROGRAMMATŪRAS PIRMKODS

package com.kvd.alarmclock;

import android.app.ListActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;

import android.view.Window;

import android.app.AlertDialog;

import android.content.Context;

import android.content.DialogInterface;

import android.content.DialogInterface.OnClickListener;

public class AlarmListActivity extends ListActivity {

private AlarmListAdapter mAdapter;

private AlarmDBHelper dbHelper = new AlarmDBHelper(this);

private Context mContext;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

mContext = this;

requestWindowFeature(Window.FEATURE\_ACTION\_BAR);

//setContentView(R.layout.activity\_alarm\_list);

mAdapter = new AlarmListAdapter(this, dbHelper.getAlarms());

setListAdapter(mAdapter);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.alarm\_list, menu);

return true;

}//creates buttons in action bar

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.action\_add\_new\_alarm: {

startAlarmDetailsActivity(-1);

break;

}

}

return super.onOptionsItemSelected(item);

}//works with buttons in action bar,

//calls those functions or views, whatever i need

@Override

protected void onActivityResult(int requestCode, int resultCode, Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if (resultCode == RESULT\_OK) {

mAdapter.setAlarms(dbHelper.getAlarms());

mAdapter.notifyDataSetChanged();

}

}

public void setAlarmEnabled(long id, boolean isEnabled) {

AlarmManagerHelper.cancelAlarms(this);

AlarmModel model = dbHelper.getAlarm(id);

model.isEnabled = isEnabled;

dbHelper.updateAlarm(model);

AlarmManagerHelper.setAlarms(this);

}

public void startAlarmDetailsActivity(long id) {

Intent intent = new Intent(this, AlarmDetailsActivity.class);

intent.putExtra("id", id);

startActivityForResult(intent, 0);

}

public void deleteAlarm(long id) {

final long alarmId = id;

AlertDialog.Builder builder = new AlertDialog.Builder(this);

builder.setMessage("Please confirm")

.setTitle("Delete set?")

.setCancelable(true)

.setNegativeButton("Cancel", null)

.setPositiveButton("Ok", new OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

//Cancel Alarms

AlarmManagerHelper.cancelAlarms(mContext);

//Delete alarm from database by its id

dbHelper.deleteAlarm(alarmId);

//Showing the new list of alarms

mAdapter.setAlarms(dbHelper.getAlarms());

//and here the alarm dissapears from view:

mAdapter.notifyDataSetChanged();

}

}).show();

}

}

package com.trigg.alarmclock;

import android.net.Uri;

public class AlarmModel {

//this is for saving alarm clock data

public static final int SUNDAY = 0;

public static final int MONDAY = 1;

public static final int TUESDAY = 2;

public static final int WEDNESDAY = 3;

public static final int THURSDAY = 4;

public static final int FRDIAY = 5;

public static final int SATURDAY = 6;

public long id = -1;

public int timeHour;

public int timeMinute;

private boolean repeatingDays[];

public boolean repeatWeekly;

public Uri alarmTone;

public String name;

public boolean isEnabled;

public AlarmModel() {

repeatingDays = new boolean[7];

}

public void setRepeatingDay(int dayOfWeek, boolean value) {

repeatingDays[dayOfWeek] = value;

}

public boolean getRepeatingDay(int dayOfWeek) {

return repeatingDays[dayOfWeek];

}

}

package com.trigg.alarmclock;

import android.app.Service;

import android.content.Intent;

import android.os.IBinder;

public class AlarmService extends Service {

public static String TAG = AlarmService.class.getSimpleName();

@Override

public IBinder onBind(Intent intent) {

return null;

}

@Override

public int onStartCommand(Intent intent, int flags, int startId) {

Intent alarmIntent = new Intent(getBaseContext(), AlarmScreen.class);

alarmIntent.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

alarmIntent.putExtras(intent);

getApplication().startActivity(alarmIntent);

AlarmManagerHelper.setAlarms(this);

return super.onStartCommand(intent, flags, startId);

}

}

package com.trigg.alarmclock;

import java.util.ArrayList;

import java.util.List;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.net.Uri;

import com.trigg.alarmclock.AlarmContract.Alarm;

public class AlarmDBHelper extends SQLiteOpenHelper {

public static final int DATABASE\_VERSION = 1;

public static final String DATABASE\_NAME = "alarmclock.db";

private static final String SQL\_CREATE\_ALARM = "CREATE TABLE " + Alarm.TABLE\_NAME + " (" +

Alarm.\_ID + " INTEGER PRIMARY KEY AUTOINCREMENT," +

Alarm.COLUMN\_NAME\_ALARM\_NAME + " TEXT," +

Alarm.COLUMN\_NAME\_ALARM\_TIME\_HOUR + " INTEGER," +

Alarm.COLUMN\_NAME\_ALARM\_TIME\_MINUTE + " INTEGER," +

Alarm.COLUMN\_NAME\_ALARM\_REPEAT\_DAYS + " TEXT," +

Alarm.COLUMN\_NAME\_ALARM\_REPEAT\_WEEKLY + " BOOLEAN," +

Alarm.COLUMN\_NAME\_ALARM\_TONE + " TEXT," +

Alarm.COLUMN\_NAME\_ALARM\_ENABLED + " BOOLEAN" +

" )";

private static final String SQL\_DELETE\_ALARM =

"DROP TABLE IF EXISTS " + Alarm.TABLE\_NAME;

public AlarmDBHelper(Context context) {

super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

}

@Override

public void onCreate(SQLiteDatabase db) {

db.execSQL(SQL\_CREATE\_ALARM);

}

@Override

public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

db.execSQL(SQL\_DELETE\_ALARM);

onCreate(db);

}

private AlarmModel populateModel(Cursor c) {

AlarmModel model = new AlarmModel();

model.id = c.getLong(c.getColumnIndex(Alarm.\_ID));

model.name = c.getString(c.getColumnIndex(Alarm.COLUMN\_NAME\_ALARM\_NAME));

model.timeHour = c.getInt(c.getColumnIndex(Alarm.COLUMN\_NAME\_ALARM\_TIME\_HOUR));

model.timeMinute = c.getInt(c.getColumnIndex(Alarm.COLUMN\_NAME\_ALARM\_TIME\_MINUTE));

model.repeatWeekly = c.getInt(c.getColumnIndex(Alarm.COLUMN\_NAME\_ALARM\_REPEAT\_WEEKLY)) == 0 ? false : true;

model.alarmTone = c.getString(c.getColumnIndex(Alarm.COLUMN\_NAME\_ALARM\_TONE)) != "" ? Uri.parse(c.getString(c.getColumnIndex(Alarm.COLUMN\_NAME\_ALARM\_TONE))) : null;

model.isEnabled = c.getInt(c.getColumnIndex(Alarm.COLUMN\_NAME\_ALARM\_ENABLED)) == 0 ? false : true;

String[] repeatingDays = c.getString(c.getColumnIndex(Alarm.COLUMN\_NAME\_ALARM\_REPEAT\_DAYS)).split(",");

for (int i = 0; i < repeatingDays.length; ++i) {

model.setRepeatingDay(i, repeatingDays[i].equals("false") ? false : true);

}

return model;

}//puts data into model from database, after that returns model

private ContentValues populateContent(AlarmModel model) {

ContentValues values = new ContentValues();

values.put(Alarm.COLUMN\_NAME\_ALARM\_NAME, model.name);

values.put(Alarm.COLUMN\_NAME\_ALARM\_TIME\_HOUR, model.timeHour);

values.put(Alarm.COLUMN\_NAME\_ALARM\_TIME\_MINUTE, model.timeMinute);

values.put(Alarm.COLUMN\_NAME\_ALARM\_REPEAT\_WEEKLY, model.repeatWeekly);

values.put(Alarm.COLUMN\_NAME\_ALARM\_TONE, model.alarmTone != null ? model.alarmTone.toString() : "");

values.put(Alarm.COLUMN\_NAME\_ALARM\_ENABLED, model.isEnabled);

String repeatingDays = "";

for (int i = 0; i < 7; ++i) {

repeatingDays += model.getRepeatingDay(i) + ",";

}

values.put(Alarm.COLUMN\_NAME\_ALARM\_REPEAT\_DAYS, repeatingDays);

return values;

}

public long createAlarm(AlarmModel model) {

ContentValues values = populateContent(model);

return getWritableDatabase().insert(Alarm.TABLE\_NAME, null, values);

}//create new alarm

public long updateAlarm(AlarmModel model) {

ContentValues values = populateContent(model);

return getWritableDatabase().update(Alarm.TABLE\_NAME, values, Alarm.\_ID + " = ?", new String[] { String.valueOf(model.id) });

}//update existing alarm

public AlarmModel getAlarm(long id) {

SQLiteDatabase db = this.getReadableDatabase();

String select = "SELECT \* FROM " + Alarm.TABLE\_NAME + " WHERE " + Alarm.\_ID + " = " + id;

Cursor c = db.rawQuery(select, null);

if (c.moveToNext()) {

return populateModel(c);

}

return null;

}

public List<AlarmModel> getAlarms() {

SQLiteDatabase db = this.getReadableDatabase();

String select = "SELECT \* FROM " + Alarm.TABLE\_NAME;

Cursor c = db.rawQuery(select, null);

List<AlarmModel> alarmList = new ArrayList<AlarmModel>();

while (c.moveToNext()) {

alarmList.add(populateModel(c));

}

if (!alarmList.isEmpty()) {

return alarmList;

}

return null;

}//gets all alarms to create main program view later - list of all alarms

public int deleteAlarm(long id) {

return getWritableDatabase().delete(Alarm.TABLE\_NAME, Alarm.\_ID + " = ?", new String[] { String.valueOf(id) });

}//delete selected alarm

}

Kvalifikācijas darbs „Android lietotne – uzlabots modinātājs” izstrādāts Latvijas Universitātes Datorikas fakultātē.

Ar savu parakstu apliecinu, ka darbs izstrādāts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Andris Niedrītis \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ **02.06.2014**.

Rekomendēju darbu aizstāvēšanai

Darba vadītājs**: *Bc.datz., Gundars Lisovskis*** \_\_\_\_\_\_\_\_\_\_\_\_\_ **02.06.2014.**

Recenzents: ***M.dat, Raimonds Simanovskis***

Darbs iesniegts 02.06.2014.

Kvalifikācijas darbu pārbaudījumu komisijas sekretārs: *Imants Gorbāns* \_\_\_\_\_\_\_\_\_\_

Darbs aizstāvēts kvalifikācijas darbu pārbaudījuma komisijas sēdē

\_\_\_.06.2014. prot. Nr. \_\_\_\_\_\_\_\_\_\_

Komisijas sekretārs(-e): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_