**Profesionālās izglītības kompetences centrs**

**„Rīgas Valsts tehnikums”**

Izglītības programma: Programmēšana

**KVALIFIKĀCIJAS DARBS**

**Mobila lietojumprogramma sporta nodarbības procesa kontrolei**

Paskaidrojošais raksts 64 lpp.

Audzēknis: Antons Cvetkovs

Vadītājs: Normunds Pauders

**Rīga**

**2022**

**ANOTĀCIAJA**

Kvalifikācijas darba ietvaros tika izstrādāta mobila lietojumprogrammā sporta nodarbības procesa kontrolei. Darba autors ir Rīgas Valsts tehnikuma audzēknis Antons Cvetkovs. Darba vadītājs ir Rīgas Valsts tehnikuma skolotājs Normunds Pauders.

Kvalifikācijas darbā izstrādātā lietotne ir sporta nodarbību kontroles bāze, kas sevī iekļauj treniņu programmas redaktoru un vingrinājuma datu bāzi. Ir arī iespēja apskatīt un izpildīt jau iepriekš izveidotus treniņu programmas. Kvalifikācijas darba sistēmā galvenais uzsvars tiek likts uz lietotāja rezultātu saglabāšanai, tāpēc lietojumprogrammu var izmantot tikai piereģistrēts lietotājs. Sistēma tika izstrādāta ar programmēšanas valodu Kotlin un datu bāzes servisu Google Firebase.

Kvalifikācijas darba paskaidrojošais raksts satur sevī ievadu, uzdevuma nostādni, prasību specifikāciju, uzdevuma risināšanas līdzekļu izvēles pamatojumu, programmatūras produkta modelēšanas un projektēšanas aprakstu, datu struktūru aprakstu, lietotāja ceļvedi, nobeigumu un pielikums. Ievadā ir aprakstītas aktuālās veselības nozares problēmas un pamatota vajadzība pēc portatīvas mobilas fitnesa lietojumprogramma. Uzdevuma nostādne ir aprakstīts kvalifikācijas darba galvenais izveidošanas mērķis un tā uzdevumi. Prasību specifikācijā ir aprakstītas sistēmas funkcionālās un nefunkcionālās prasības, kā arī aplūkojama sistēmas izejas un ieejas informācija – dati, ko lietotājs ievada un dati, kuri tiek tam izvadīti. Uzdevuma risināšanas līdzekļu izvēles pamatojumā ir aprakstīts, kāda programmēšanas valoda, teksta redaktors, relāciju datu bāžu vadības sistēma un failu apmaiņas programma tika izmantota sistēmas izstrādē. Programmatūras produkta modelēšanas un projektēšanas aprakstā ir apskatāmas datu plūsmu diagrammas sistēmas funkcionēšanas attēlošanai, ER diagrammas datu bāzes uzbūves un tās pamatprincipu izprašanai, kā arī detalizēti aprakstīta sistēmas arhitektūra. Datu struktūru apraksts satur sevī visu tabulu struktūru un to aprakstu, kā arī tabulu relāciju shēmu. Lietotāja ceļvedis detalizēti attēlo sistēmas informācijas vizuālo izkārtojumu un paskaidro kā pareizi lietot sistēmu. Nobeigumā ir aprakstīts, kas tik izveidots. Pielikums satur datu bāzes klāšu diagrammu.

Kopumā kvalifikācijas darba apjoms ir 64. lpp., kurā ietilpst 65 attēli, 9 tabulas un 1 pielikums.

**ANNOTATION**

Within the framework of the qualification work there was developed a mobile application for controlling the process of workouts. The Author is Riga State Technical School student Antons Cvetkovs. The supervisor is Riga State Technical School teacher Normunds Pauders.

Qualification work developed system is a mobile application for Android for sports activities, which includes a workout editor, creator and exercises database. It is also possible to view and do premade workouts. In the qualification work system, the main emphasis is on saving users results, so the application can only be used by a registered one. The system was developed using Kotlin programming language and the Google Firebase database service.

Qualification work explanatory consists of introduction, task formulation, the requirements of the specification, task execution tool explanation, program product modeling and projecting description, data structure description, user guide, conclusion and attachments. Introduction describe what is the main problem in forestry and why this application is needed. Task formulation describes the most important qualification work tasks. The requirements of the specification describe functional and non-functional actions, it also describes the input and output data. Task execution tool explanation describes the tools that were used in the qualification work creation process. Program product modeling and projecting description includes data flow diagrams and ER diagrams, data structure description tells everything about qualification work database structure and explains every field in it. User guide describes how to use system and how to use it correctly. Conclusion consists of description of what was achieved and what was discovered in the process of development. Attachments includes database class diagram.

Overall, qualification work consists of 64 pages, which includes 65 images, 9 tables and 1 attachment.

**SATURS**

**IEVADS**5

**1. UZDEVUMU NOSTĀDNE**6

**2. PRASĪBU SPECIFIKĀCIJA** 7

2.1. Ieejas un izeja informācijas apraksts7

2.1.1. Ieejas informācijas apraksts7

2.1.2. Izejas informācijas apraksts7

2.2. Funkcionālas prasības8

2.3. Nefunkcionālas prasības10

2.4. Skices11

**3. UZDEUMU RISINĀŠANAS IZVĒLES PAMATOJUMS**12

**4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA**13

4.1. Sistēmas struktūras modelis 13

4.1.1. Sistēmas arhitektūra13

4.1.2. Sistēmas ER modelis15

4.2. Funkcionālas sistēmas modelis17

4.2.1. Datu plūsmu modelis17

**5. DATU STRUKTŪRU APRAKSTS**33

**6. LIETOTĀJA CEĻVEDIS**36

6.1. Sistēmas prasības aparatūrai un programmatūrai36

6.2. Sistēmas instalācija un palaišana37

6.3. Programmas apraksts39

6.4. Testa piemērs56

**NOBEIGUMS**61

**INFORMĀCIJAS AVOTI**62

**PIELIKUMI**63

1. pielikums64

**IEVADS**

Cilvēka veselība ir nenovērtējama dāvana, kas cilvēkam dota. Diemžēl mēs bieži pret to izturamies vieglprātīgi un bezatbildīgi, par to nerūpējamies, pat nedomājam par to, sevišķi kad dzīve iet pašā plaukumā! Mūsdienas cilvēkam ir īpaši grūti atbalstīt to – pasīvais dzīvesveids, cukura, sojas un citu piedevu pievienošanā lielākā daļa no ēdieniem, pārtikas ražošana industriālā lielumā, nepareiza dieta un dzīve pilsētā – tas viss izraisa problēmas ar veselību.

Mūsdienas, cilvēki tērē lielāku daļu no savas dienas uz darbu vai mācībām un lai ietaupīt savu laiku, viņi pasuta vai nopērk ēdienu kafejnīcās, veikalos un, kas ir viss sliktākais variants, ātrās ēdināšanas restorānos. Cilvēks nezin kā bija gatavots viņa ēdiens un, kas bija pievienots tā. Lai ēdiens likās garšīgāks klientam, tā pievieno taukus un cukuru, apvienojot to ar pasīvu dzīvesveidu – cilvēkam izraisās problēmas ar lieko svaru.

Liekais svars ir viena no Zemes lielākām problēmām. Pasaulē vairāk nekā 1,4 miljardiem pieaugušo, kas ir vecāki par 20 gadiem, ir konstatēts liekais svars. Turpretim vairāk nekā 200 miljoni vīriešu un 300 miljonu sieviešu cieš no aptaukošanās un tās izraisītajām veselības problēmām. Saskaņā ar Pasaules veselības organizācijas datiem, vairāk nekā 40 miljonu bērnu, vecumā līdz 5 gadiem, ir aptaukojušies.

Pārtikas ražošana industriālā lielumā un liela konkurence liek lieliem ražotajiem padarīt savu produktu lētāko un vieglāk ražojumu. Tāpēc vairāki pārtikas produkti tiek ražoti nedabiska veida, piemēram gaļa vai piena produkti, kas bija izgatavoti no sojas. Piemēram, tādēļ, vīriešiem ir kaitīgs ne tikai piens, kas bija pagatavots no sojas, bet arī vienkāršs, “naturāli” izgatavots, jo piens, kas tiek pārdots veikalos ir piepildīts ar sieviešu hormoniem un tajā gandrīz nav vīriešu hormonu, tas izraisa problēmu ar testosterona līmeni vīriešu organisma. Mūsdienas, vīriešiem līdz 22. gada vecumam, testosterona līmenis ir zemāks nekā 33. gadīgiem 2007. gadā.

Viens no variantiem, lai palīdzēt risināt visas tas problēmas ir pareizas sporta nodarbības. Sporta nodarbības palīdz sadedzināt kalorijas, kas savukārt, apvienojot to ar pareizu dietu un kalorijas deficītu, palīdz dedzināt taukus. Kā arī, treniņi palielina tostestorana līmeni vīriešu organismā un padara cilvēka ķermenī spēcīgāku. Nodarbības ar sportu motivē cilvēku rūpēties par savu ķermeni un veselibu. Tāpēc, es uzskatu, ka mūsdienas cilvēkam ir obligāti jānodarbojas ar sportu.

**1. UZDEVUMU NOSTĀDNE**

**Kvalifikācijas darba uzdevums** ir izveidot ērtu mobilo lietojumprogramma, kas ļaus lietotājam izveidot savus treniņu programmas, saglabāt izpildīto treniņa rezultātu un kas atbildēs mūsdienas dizaina prasībām.

**Kvalifikācijas darba funkcionalitātes:**

* lietotāja datu – vārda, auguma un svara – pievienošana, rediģēšana, apstrāde un dzēšana;
* lietotāja profila izveide un dzēšana;
* treniņu programmas izveide, rediģēšana un dzēšana;
* treniņu izpildīšana;
* treniņa un vingrinājuma rezultātu saglabāšana un dzēšana.

**2. PRASĪBU SPECIFIKĀCIJA**

**2.1. Ieejas un izeja informācijas apraksts**

***2.1.1. Ieejas informācijas apraksts***

1. Informācija par **lietotāju**, kas sastāv no:

* e-pasta, kas ir simbolu virkne, ne garāka par 254 simboliem – maksimālo iespējamo e-pasta garumu, pēc *Internet Engineering Task Force* *(IETF)* standartiem;
* lietotāja vārda, kas ir simbolu virkne;
* lietotāja auguma, kas ir simbolu virkne, ne garāka par 5 simboliem;
* lietotāja svara, kas ir simbolu virkne, ne garāka par 5 simboliem.

1. Informācija par **lietotāja izveidotiem treniņiem**, kas sastāv no:

* nosaukums, kas ir simbolu virkne;
* treniņi, kas ir saraksts, kas sastāv no vingrinājumiem, kurus izvēles lietotājas.

1. Informācija par **vingrinājuma, izpildītiem piegājieniem**, kas sastāv no:

* atkārtojumā daudzuma, kas ir ciparu virkne;
* inventāra svaru, ar kuru bija izpildīts vingrinājums.

***2.1.2. Izejas informācijas apraksts***

Izejas dati ir no datu bāzes izgūtie dati, kā atbilde uz lietotāja pieprasījumu.

1. Informācija par **lietotāju**, kas sastāv no:

* lietotāja vārds;
* lietotāja svars;
* lietotāja augums;

1. Informācija par **standartiem, iepriekš izveidotiem treniņiem**, kas sastāv no:

* treniņa nosaukums;
* treniņa apraksts;
* treniņa vingrinājuma daudzums;
* treniņa vingrinājumu saraksts.

1. Informācija par **lietotāja treniņiem**, kas sastāv no:

* lietotāja treniņa nosaukums;
* lietotāja treniņa vingrinājuma daudzums;
* lietotāja treniņa vingrinājumu saraksts.

1. Informācija par **vingrinājumiem**, kas sastāv no:

* vingrinājuma nosaukums;
* vingrinājuma bilde;
* vingrinājuma inventārs, kas ir vajadzīgs vingrinājuma izpildei;
* vingrinājuma muskuļu grupa, kas ir trenēta, pildot vingrinājumu;
* vingrinājuma apraksts, jeb instrukcija, ka pareizi izpildīt to.

1. Informācijā par **uzsāktiem/izpildītiem treniņiem**, kas sastāv no:

* lietotāja izpildīto treniņu daudzumus;
* lietotāja uzsāktu treniņu daudzums;
* uzsākto treniņu nosaukumu;
* uzsākto treniņu izpildes datumu;
* uzsākto treniņu statusu (pabeigts/nepabeigts).

1. Meklēšanas rezultātu atspoguļojums. Aplikācijā būs iespēja meklēt vingrinājums pēc ķermeņa daļas uz kuru ir lielāka slodze. Šādā veidā tiek nodrošināta iespēja atrast vispiemērotākos vingrinājumus.

**2.2. Funkcionālās prasības**

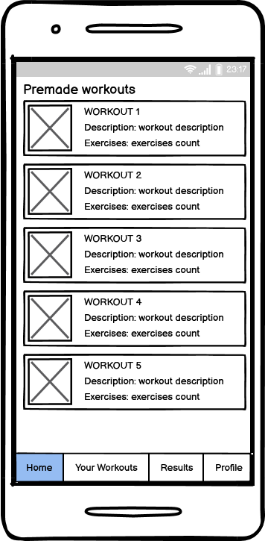
1. Lietotāja reģistrācija:
   1. lietotnei jānodrošina viegli lietojamu grafisko lietotāja dizainu (GUI), lai lietotājs varētu reģistrēties, kad lietojumprogramma tiek palaista.;
   2. lietotnei jānodrošina profila izveidi ar Google kontu;
   3. lietotnei ir jāinformē lietotāju, ja tam nav interneta savienojuma un, tāpēc viņš nevar piereģistrēties;
   4. lietotāja datu pievienošanu – vārds, svars un augums;
   5. atgādinājuma par treniņu pievienošana;
   6. lietotnei ir jāizveido lietotāja unikālo atslēgu (ID).
2. Lietotāja pieslēgšana savam profilam:
   1. lietotnei jānodrošina viegli lietojamu GUI, lai lietotājs varētu reģistrēties, kad lietojumprogramma tiek palaista;
   2. lietotnei jānodrošina pieslēgšanu ar Google kontu;
   3. lietotnei ir jāinformē lietotāju, ja tam nav interneta savienojuma un, tāpēc viņš nevar pieslēgties.
3. Lietotāja profila iespējas:
   1. lietotāja datu – vārda, svara un auguma – maiņa;
   2. profila apskatīšana;
   3. rezultātu dzēšana;
   4. profila dzēšana;
   5. iziešana no profila;
   6. atgādinājuma maiņa.
4. Lietotāja treniņi:
   1. lietotnei ir jānodrošina intuitīvo GUI, lai lietotājam būtu iespējā viegli izveidot treniņu;
      1. lietotnei ir jāaizliedz lietotājam izveidot treniņu bez vingrinājumiem;
      2. gadījumā, kad lietotājs neievada treniņa nosaukumu, lietotne pievieno nosaukumu “WORKOUT” ar lietotāja treniņa kārtas numuru;
   2. informācijas par treniņu atspoguļojums;
   3. lietotnei ir jānodrošina intuitīvo GUI, lai lietotājam būtu iespējā rediģēt izveidoto treniņu;
   4. lietotāja treniņa izpildi;
      1. taimera atspoguļojumu starp vingrinājumiem un tā piegājieniem;
      2. vingrinājuma piegājienu pievainošana izpildes laikā;
      3. atkārtojumu un inventāra svara pievainošana;
      4. informācija par pildīto vingrinājumu atspoguļojums.
5. Standarti treniņi:
   1. informācijas par treniņu atspoguļojums;
   2. treniņu izpilde:
      1. taimera atspoguļojums starp vingrinājumiem un tā piegājieniem;
      2. vingrinājuma piegājienu pievainošanu treniņa izpildes laikā;
      3. atkārtojumā un inventāra svara pievainošana;
      4. informācija par pildīto vingrinājumu atspoguļojums.
6. Rezultāti:
   1. lietotāja izpildīto treniņu atspoguļojums;
      1. lietotāja izpildīto treniņu datu – izpildes datuma, treniņa nosaukuma un statusa – atspoguļojums;
   2. izpildīto un uzsāktu treniņa daudzumu atspoguļojums;

**2.3. Nefunkcionālās prasības**

1. Prasības procesam:
   1. lietotnes datu bāzei jābūt izstrādātai izmantojot Google Firebase servisu;
   2. lietotnei jābūt izstrādātai uz Kotlin – 1.6.0. versijas – gan lietotāja, gan servera pusē;
   3. lietotnei jābūt ātram ielādēs laikam un tā nedrīkst pārslogot mobilas ierīces procesoru.
2. Prasības produktam:
   1. lietotāju saskarei ar lietotni jānotiek angļu valodā;
   2. lietotnes atbildes laikam izpildot jebkuru sarežģītības pieejamo operāciju jābūt līdz 3 sekundēm no pieprasījuma saņemšanas brīža;
   3. lietotnei jābūt savietojamai uz visiem ekrānu izmēriem un to izšķirtspējām.
3. Prasības lietotāja interfeisam:
   1. lietotnes pamatkrāsām jābūt – melnām, tumši pelēkam un baltam;
   2. visu logu galvenajiem virsrakstiem un tekstam jābūt vienotā stilā;
   3. kļūdu paziņojumiem jābūt vienotā stilā;
   4. lietotnei jābūt ērti pārskatāmai un lietojamai uz jebkuras mobilas ierīces;
   5. lietotnei jābūt vieglam, ērtām un mūsdienīgam dizainam.

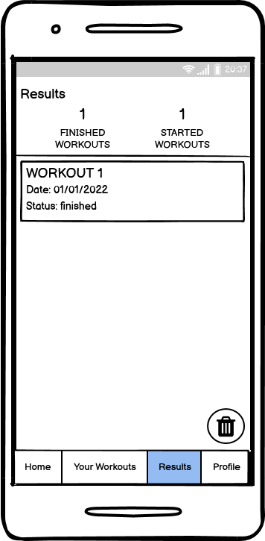
**2.4. Skices**

**Galvenā loga skice**

Galvenā loga var redzēt loga nosaukumu “Premade workouts” un standarta treniņa sarakstu, ar treniņa bildi, nosaukumu, muskuļu grupu, kas ir trenēta un vingrinājuma skaitu, apakšā atrodas sekciju izvēlne.

2.1. att. Galvenā loga skice

**Rezultāta loga skice**

Rezultātu loga lietotājas var aplūkot uzsākto un pabeigto treniņa daudzumu, uzsāktu treniņa sarakstu ar tā nosaukumu, izpildes datumu un statusu (pabeigts vai nepabeigts), un labā apakšā stūrī dzēšanas pogu.

2.2. att. Rezultātu loga skice

**3. UZDEUMU RISINĀŠANAS IZVĒLES PAMATOJUMS**

Sistēma tika būvēta izmantojot vairākus populārus, publiskus ietvarus. Projekta priekšgala interfeiss balstās uz Kotlin programmēšanas valodas. Sistēmas izstrādei bija izvēlēta Kotlin valoda, jo 2019. gadā Google nosauca to par primāro programmēšana valodu Android izstrādē un Kotlin ir paredzēts, lai pilnībā sadarbotos ar Java, tāpēc visas Java bibliotēkas strādā ar to. Projektā izmantos Kotlin 1.6.20.

Ka izstrādes platforma tika izvēlēts Android, jo, apmērām, 71.45% no visam mobilo ierīču pasaulē ir taisīti uz tā.

Datu bāzes izstrādei un failu glābšanai tika izmantots Firebase serviss. Firebase ir Google izstrādāta platforma mobilo un tīmekļa lietojumprogrammu izveidei. Firebase ietver sevī vairākus rīkus – Firebase Authentication, Cloud Database, Realtime Datbase un citus – kas atvieglo sistēmās izstrādi. Visiem Firebase lietotājiem Google, par brīvu, piedāvā izmantot savus serverus visa pasaulē.

Par lietotāja konta reģistrēšanu sistēmā atbildēs Firebase Authentication rīks. Tas ļauj lietotājam reģistrēties sistēma izmantojot savu Google kontu. Pats rīks saglāba lietotāja e-pastu un izveido viņam unikālu atslēgu(ID).

Failu glabāšanai tika izmantots Firebase Storage un pašai datu bāzei bija izmantots Firebase Realtime Database. Firebase Realtime Database ir NoSQL datu bāze. Dati glābās JSON failā un reāllaikā sinhronizējas ar visiem pieslēgtiem lietotājiem.

Sistēmas izstrādei tika izmantota Android Studio integrēta izstrādes vide (IDE), jo tā ir izstrādāta Android lietotņu projektēšanai.

**4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA**

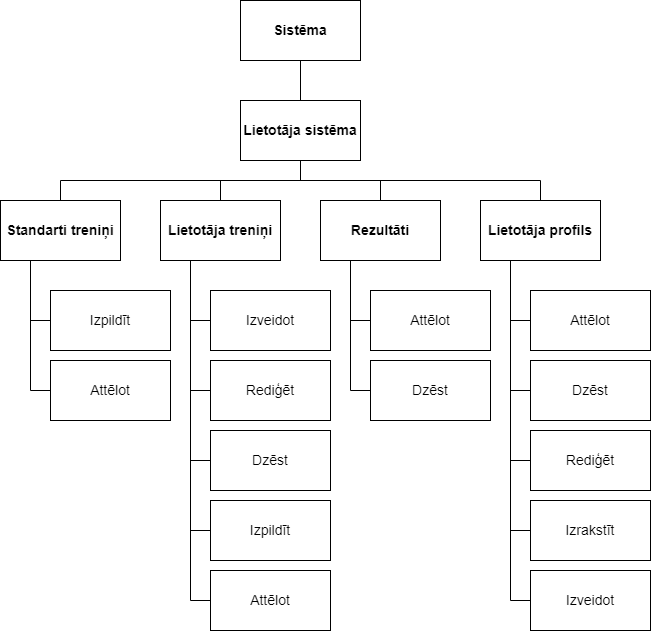
**4.1. Sistēmas struktūras modelis**

***4.1.1. Sistēmas arhitektūra***

Sistēmā ir realizēta tikai viena loma – lietotājs. Lietotājam ir iespēja pildīt standartu treniņa programmas, ka arī veidot savas, pievienot savus vingrinājums un apskatīt tos, kas jau ir iekļautas lietotnē, apskatīt savus rezultātus un rediģēt informāciju par sevi – augums, svars un vārds.

Sistēmā ir četri moduli.

* Modulis “Standarti treniņi”:
  + lietotājam ir iespēja apskatīt un izpildīt treniņus.
* Modulis “Lietotāja treniņi”:
  + lietotājam ir iespēja izveidot, rediģēt, dzēst, izpildīt un apskatīt lietotāja treniņus.
* Modulis “Rezultāti”:
  + lietotājam ir iespēja apskatīt savus treniņa rezultātus un dzēst tos.
* Modulis “Lietotāja profils”:
  + lietotājam ir iespēja izveidot, apskatīt, rediģēt, dzēst savu profilu un izrakstīties no tā.

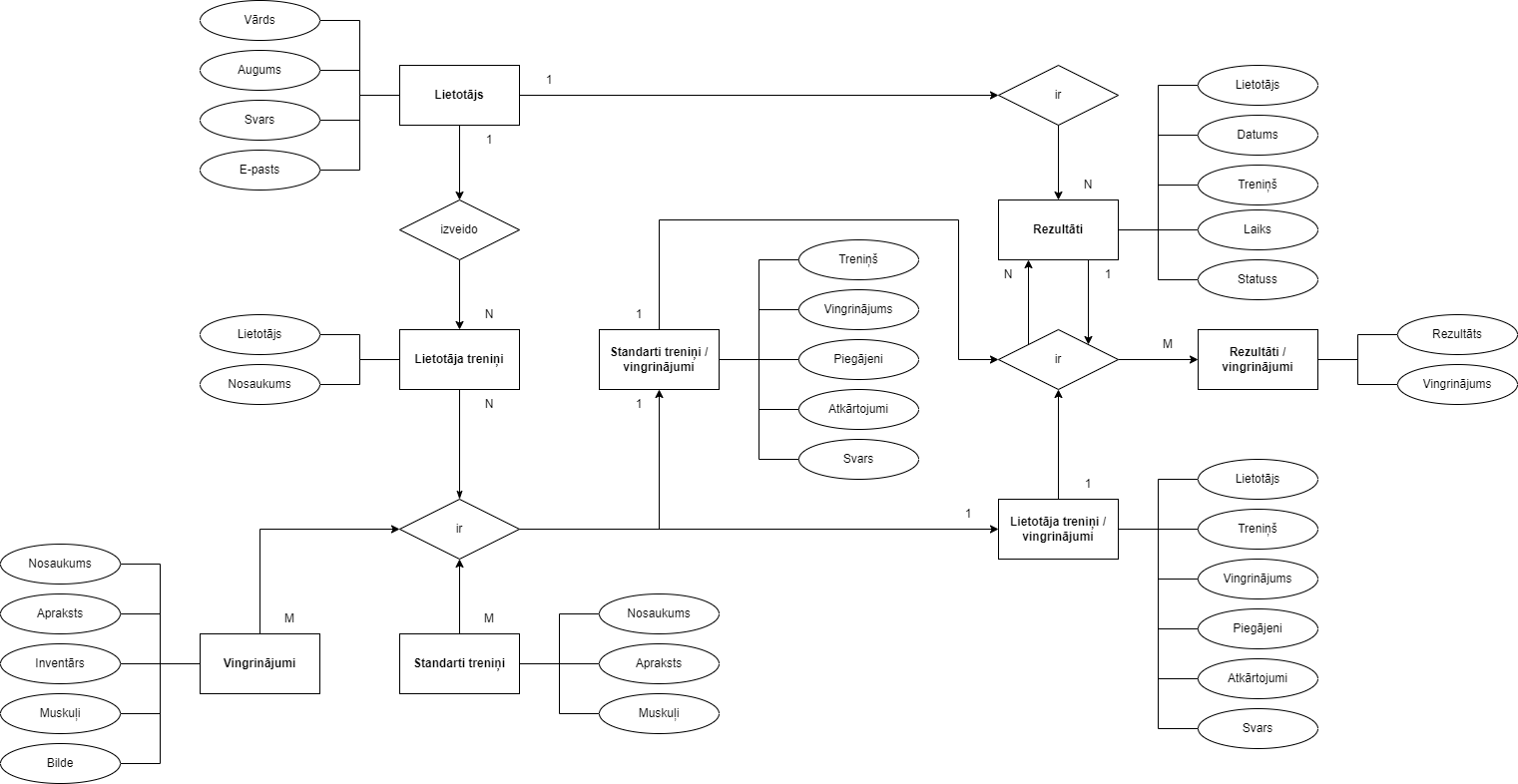
4.1. att. Sistēmas arhitektūra

***4.1.2. Sistēmas ER modelis***

Datu bāzes projektēšanā datu kopu un saišu starp tām attēlošanai tika lietota realitāšu-saišu diagramma, kas sastāv no divu veidu objektiem – entītijām (loģiskais objekts) un relācijām (attiecības). ER modelis sastāv no 5 entītijām un 3 starp tabulām, kas atspoguļo datu aprīti sistēmā.

* **“Lietotājs”** – uzskaita sistēmas lietotājus. Tās atribūtu kopums sevī ietver e-pastu, vārdu, augumu, svaru un vecumu.
* **“Lietotāja treniņi”** – uzskaita informāciju par treniņiem. Tās atribūtu kopums sevī ietver lietotāju, muskuļu grupu un nosaukumu.
* **“Standarti treniņi”** – uzskaita informāciju par treniņiem. Tās atribūtu kopums sevī ietver nosaukumu, muskuļu grupu un aprakstu.
* **“Vingrinājumi”** – uzskaita informāciju par vingrinājumiem. Tās atribūtu kopums sevī ietver nosaukumu, aprakstu, inventāru, muskuļu grupu un bildi ar vingrinājuma piemēru.
* **“Standarti treniņi / vingrinājumi”** – uzskaita informāciju par vingrinājumiem, kas bija izpildīti treniņā. Tās atribūtu kopums sevī ietver treniņu, vingrinājumus, piegājienus, atkārtojumus, svaru (inventārs).
* **“Lietotāja treniņi / vingrinājumi”** – uzskaita informāciju par vingrinājumiem, kas bija izpildīti lietotāja treniņā. Tās atribūtu kopums sevī ietver lietotāju, treniņu, vingrinājumus, piegājienus, atkārtojumus, svaru (inventārs).
* **“Rezultāti”** – uzskaita treniņu rezultātus. Tās atribūtu kopums sevī ietver treniņu, datumu (kad bija izpildīts treniņš) un lietotāju.
* **“Rezultāti / vingrinājums”** – uzskaita treniņa laikā izpildītos vingrinājumus. Tās atribūtu kopums sevī ietver rezultātu un vingrinājumu.

Datu bāzes relācijas uzrāda kā savstarpēji ir savienotas divas vai vairākas entītijas.

* Starp lietotājiem un lietotāja treniņiem attiecība ir viens pret daudziem, jo viens lietotājs var izveidot vairākus treniņus, bet vienu treniņu var izveidot tikai viens lietotājs.
* Starp lietotāja treniņam un vingrinājumiem ir attiecība daudzi pret daudziem, jo viena treniņā varbūt vairāki vingrinājumi, un viens vingrinājums varbūt vairākos treniņos. Veidojas tabula lietotāja treniņi/vingrinājumi.
* Starp standartam treniņam un vingrinājumiem ir attiecība daudzi pret daudziem, jo viena treniņā varbūt vairāki vingrinājumi, un viens vingrinājums varbūt vairākos treniņos. Veidojas tabula standarti treniņi/vingrinājumi.
* Starp lietotāja treniņi/vingrinājumi un rezultāti ir attiecība viens pret daudziem, jo vienam treniņam varbūt vairāki rezultāti, bet viens rezultāts varbūt vienam treniņam.
* Starp standarti treniņi/vingrinājumi un rezultāti ir attiecība viens pret daudziem, jo vienam treniņam varbūt vairāki rezultāti, bet viens rezultāts varbūt vienam treniņam.
* Starp lietotājiem un rezultātiem ir attiecība viens pret daudziem, jo viena lietotājam varbūt vairāki rezultāti, bet viens rezultāts varbūt vienām lietotājam.
* ****Starp lietotāja treniņi/vingrinājumi, standarti treniņi/vingrinājumi un rezultāti veidojas starp tabula rezultāti/vingrinājumi, kas saglāba, kādi vingrinājumi bija izpildīti treniņa laikā. Starp tabulām ir attiecība vien pret daudziem, jo viena treniņā lietotājs var izpildīt vairākus vingrinājumus.

4.2. att. Sistēmas ER modelis

**4.2. Funkcionālas sistēmas modelis**

***4.2.1 Datu plūsmu modelis***

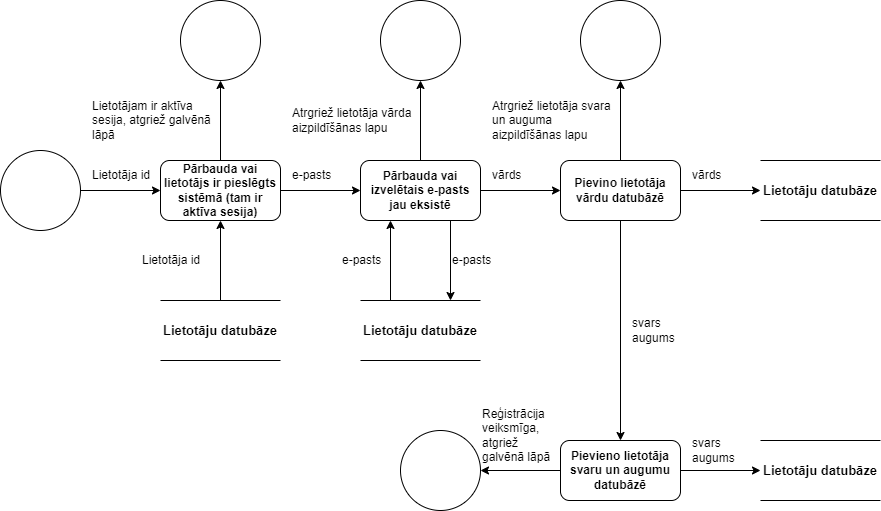
**1. Lietotāja reģistrācija**

Lietotāja profila izveide ir vieno no galvenākajam lapas funkcionalitātes sastāvdaļām, jo tālāk izmantot lietotni var tikai reģistrētie lietotāji. Tā ir izdarīts lai visi lietotāja dati, rezultāti un izveidoti treniņi glabātos serverī, tad lietotājas nepazudis savus datus arī ja viņam vajadzēs pārinstalēt lietotni. Reģistrācija notiks izmantojot Google Authentication. Lietotāja reģistrācija noris sekojoši (skat. 4.3. att.).

Vispirms tiek pārbaudīts vai lietotājs ir pieslēdzies sistēmā. Ja lietotājam ir aktīva sesija, atgriežam lietotāju sākumlapu, jo reģistrētam lietotājam nav vajadzības pēc reģistrācijas.

Ja lietotājs nav pieslēdzies un lietotājs ir izvēlējis Google kontu ar kuru viņš grib reģistrēties, tad pārbaudām vai izvelētais e-pasts jau eksistē sistēmā. Ja e-pasts neeksistē tad pievienojām to datu bāzē.

Ja e-pasts neeksistē, tad izvadām lapu, kurā lietotājam būs jāievadā savs vārds, kad tas tika ievadīts, pavienojam to datu bāzē un izvadām lietotāja lapu uz kurā lietotājam būs jāievadā savu svaru un augumu. Kad lietotājas ir ievadīs svaru un augumu, pievienojam tos sistēmā un atgriežam galveno lapu.

4.3. att. Lietotāja reģistrācijas datu plūsmu diagramma

**2. Eksistējoša lietotāja autorizācija**

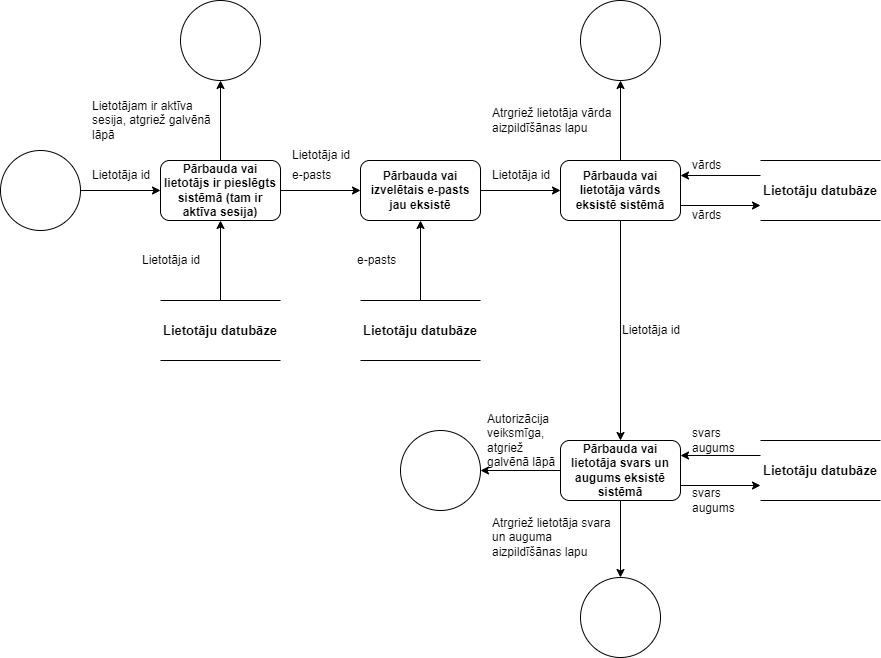
Lietotāja autorizācija sākas līdzīgi reģistrācijai. Lietotājs izvēlas savu e-pastu un sistēma pārbauda vai tas eksistē tā. Eksistējoša lietotāja autorizācija notiks sekojoši (skat. 4.4. att.).

Ja e-pasts eksistē, tad sākumā pārbauda vai lietotāja vārds eksistē sistēmā, ja nē, tad atgriežam lapu, kurā lietotājam būs jāievadā to. Kad lietotājs ievadā savu vārdu, pievienojam to datu bāzē.

Kad vārds ir pievienots, pārbaudām vai lietotājam ir piešķirti svars un augums, ja nē tad atgriežam svara un auguma ievadīšanas lapu, kad tie ir ievadīti, pievienojām tos datu bāzē. Gadījuma, kad svars un augums jau eksistē, tad atgriežam galveno lapu.

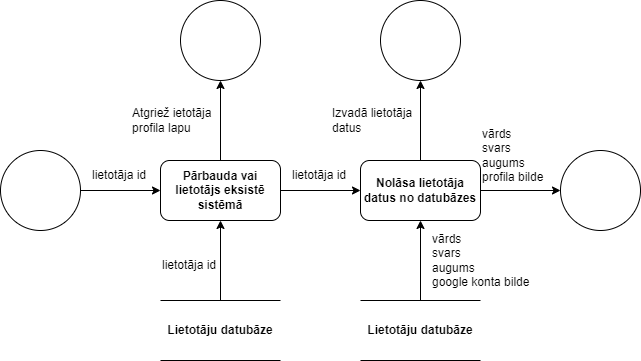
Gadījuma, kad lietotājam ir piešķirts vārds, pārbaudām vai viņam ir piešķirti svars un augums, ja nē tad atgriežam svara un auguma ievadīšanas lapu, kad tie ir ievadīti, pievienojām tos datu bāzē un atgriežam galveno lapu.

Gadījuma, kad gan vārds, gan svars ar augumu jau ir sistēmā, tad uzreiz atgriežam galveno lapu.

4.4. att. Eksistējoša lietotāja autorizācijas datu plūsmu diagramma

**3. Lietotāja profila apskatīšana**

Lietotāja profila apskatīšana notiks sekojoši (skat. 4.5. att.). Lai izvadīt lietotāja datus, sākumā sistēmā pārbauda vai lietotājs pastāv tā. Ja lietotājs pastāv, tad sistēmā dabu lietotāja vārdu, svaru, augumu un Google konta bildi no datu bāzes un izvadā tos uz ekrānā.

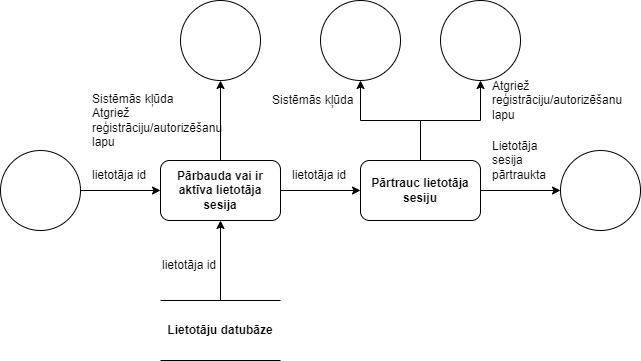
4.5. att. Lietotāja profila apskatīšanas datu plūsmu diagramma

**4. Lietotāja izrakstīšana no sistēmās (sesijas pārtraukšana)**

Lietotājam jābūt iespējai izrakstīties no sistēmās. Lietotāja izrakstīšana notiks sekojoši (skat. 4.6. att.).

Vispirms pārbaudām vai lietotājs ir pieslēdzies sistēmā (aktīva sesija). Ja lietotājam nav aktīvas sesijas un viņš var apskatīt savu profilu, tad sistēmā izvadā kļūdu, jo lietotājs nevar apskatīt savu profilu nepieslēgtos sistēmā. Atgriežam lietotāju reģistrācijas/autorizēšanas lapā.

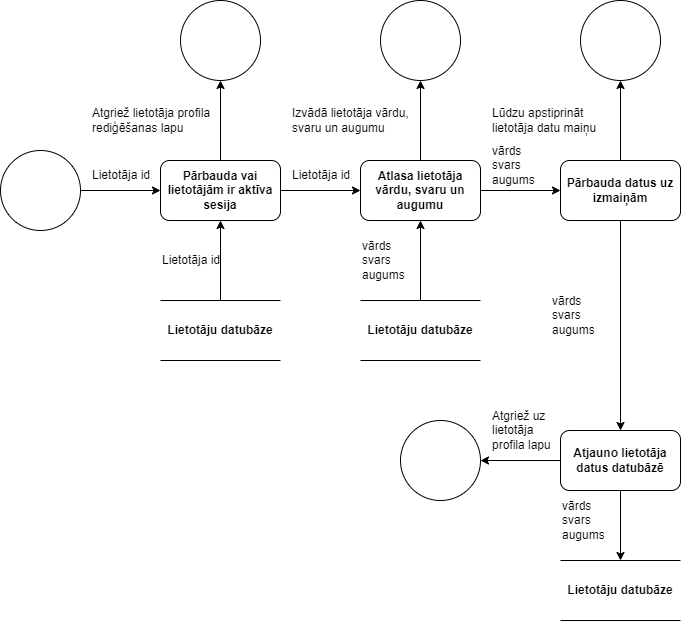
Ja lietotājas ir aktīvs – pārtraucam viņā sesiju. Ja sesijas pārtraukšanas procesa ieviešas kļūda, tiek izvadīta sistēmas kļūda. Ja kļūdas nav, atgriežam lietotāju reģistrācijas/autorizēšanas lapā. Pārtraucam lietotāja sesiju.

4.6. att. Lietotāja izrakstīšana no lietotnes datu plūsmu diagramma

**5. Lietotāja profila rediģēšana**

Lietotājam jābūt iespējai rediģēt sava profila informāciju. Lietotāja profila rediģēšana notiks sekojoši (skat. 4.7. att.).

Sākumā sistēmā pārbauda vai lietotājam ir aktīva sesija. Ja lietotājs nav autorizēts, tad izvadā sistēmas kļūdu un atgriež lietotāju uz reģistrācijas/autorizēšanas lapu. Ja kļūda nav, tad sistēmā atlasa lietotāja vārdu, svaru un augumu un izvadā tos uz ekrānā. Kad lietotājs veido izmaiņas, viņām vajag apstiprināt tās izmaiņas. Kad lietotājs apstiprina izmaiņas, sistēmā atjauno lietotāja datus datu bāzē un atgriež lietotāju sava profila lapā.

4.7. att. Lietotāja profila rediģēšanas datu plūsmu diagramma

**6. Lietotāja profila dzēšana**

Lietotājam jābūt iespējai izdzēst savu profilu. Profila dzēšana notiks sekojoši (skat. 4.8. att.).

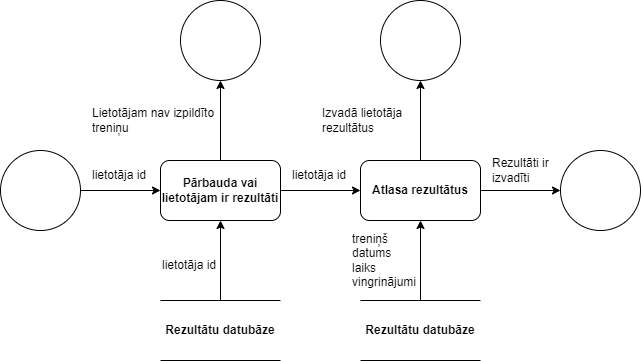
Sākumā sistēmā pārbauda vai lietotājs eksistē datu bāzē. Ja lietotājs neeksistē sistēmā, tad izvadā sistēmas kļūdu. Ja lietotājs eksistē, tad izdzēs lietotāju no datu bāzes un atgriež lietotāju uz reģistrācijas/autorizācijas lapu. Lietotāja profils ir izdzēst no sistēmās.

4.8. att. Lietotāja profila dzēšanas datu plūsmu diagramma

**7. Lietotāja rezultātu attēlošana**

Pēc katra treniņa, kuru izpildīs lietotājs, sistēma saglabās izpildīto treniņa nosaukumu un datumu, kad tā bija izpildīta. Rezultāti būs izvadīti viena sarakstā pēc izpildes datuma, sākot no pēdējā izpildīta. Lietotāja rezultātu attēlošana notiks sekojoši (skat. 4.9. att.).

Sistēma pārbauda vai lietotājam ir pabeigtie treniņi (rezultāti). Ja lietotājs nepabeidza nevienu treniņu, tad lapā būs izvadīts paziņojums, ka lietotājam nav pabeigto treniņu. Kad lietotājam ir rezultāti, tad sistēma atlasā viņa rezultātus. Rezultāti ir izvadīti.

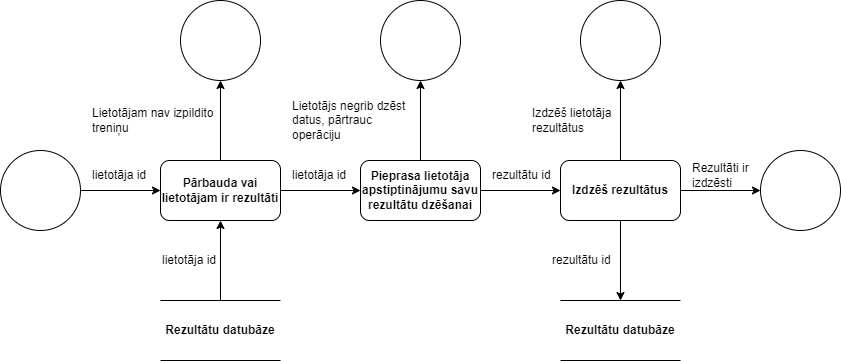
4.9. att. Lietotāja rezultātu attēlošanas datu plūsmu diagramma

**8. Lietotāja rezultātu dzēšana**

Lietotājam jābūt iespējai dzēst savus treniņa rezultātus. Lietotājam būs tikai iespēja izdzēst visus savus rezultātus vienlaicīgi. Lietotāja rezultātu dzēšana notiks sekojoši (skat. 4.10. att.).

Sakumā, sistēma pārbada vai lietotājam ir pabeigtie treniņi (rezultāti). Ja lietotājam nav pabeigto treniņu, tad logā būs izvadīts paziņojums, ka lietotājam nav pabeigto treniņu. Kad lietotājam ir rezultāti, tad sistēma atlasā viņa rezultātus.

Kad lietotājs uzspiež uz dzēšanas pogu, uz ekrāna tiek izvadīs paziņojums par to, ka ja lietotājs apstiprina rezultātu dzēšanu, tad tie tiek izdzēsti no sistēmas. Gadījumā, kad lietotājs neapstiprina rezultātu dzēšanu, operācijas tiek pārtraukta.

****4.10. Lietotāja rezultātu dzēšanas datu plūsmu diagramma

**9. Lietotāja treniņu attēlošana**

Lietotājam jābūt iespējai apskatīt savus treniņus. Visi lietotāja izveidoti treniņi būs izvadīti viena sarakstā. Lietotāja treniņu saraksta apskatīšana notiks sekojoši (skat. 4.11. att.).

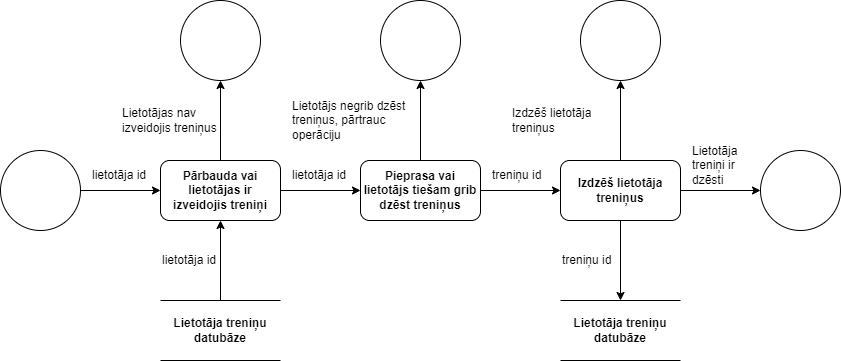
Pirmkārt, sistēma pārbauda vai lietotājam ir izveidoti treniņi. Ja tādu nav, tad uz ekrāna izvada tekstu, ka lietotājam nav izveidotu treniņu. Gadījumā, kad lietotājam ir izveidoti treniņi, sistēma izvadā tos viena sarakstā, parādot tā nosaukumu un vingrinājuma skaitu. Lietotāja treniņi ir izvadīti.

4.11. Lietotāja treniņu attēlošanas datu plūsmu diagramma

**10. Lietotāja treniņu dzēšana**

Lietotājam jābūt iespējai dzēst savus treniņus, kurus viņš ir izveidojis, gan uzreiz visus, gan katru atsevišķi. Visu lietotāja treniņu dzēšana notiks sekojoši (skat. 4.12. att.).

Pirmkārt, sistēma pārbauda vai lietotājam ir izveidoti treniņi. Ja tādu nav, tad uz ekrāna izvada tekstu, ka lietotājam nav izveidotu treniņu. Gadījumā, kad lietotājam ir izveidoti treniņi, sistēma izvadā tos viena sarakstā.

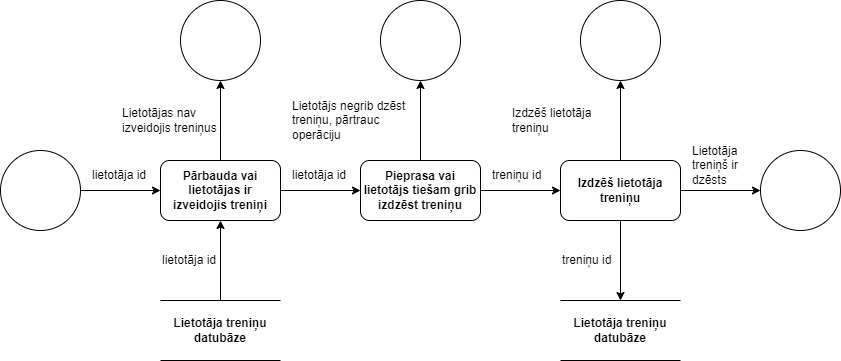
Kad lietotājs uzspiež uz dzēšanas pogu, uz ekrāna tiek izvadīts paziņojums par to, ka ja lietotājs apstiprina treniņu dzēšanu, tad tie tiek izdzēsti no sistēmas. Gadījumā, kad lietotājs neapstiprina treniņu dzēšanu, operācija tiek pārtraukta.

4.12. Visu lietotāja treniņu dzēšanas datu plūsmu diagramma

Atsevišķa lietotāja treniņa dzēšana notiks sekojoši (skat. 4.13. att.).

Pirmkārt, sistēma pārbauda vai lietotājam ir izveidoti treniņi. Ja tādu nav, tad uz ekrāna izvada tekstu, ka lietotājam nav izveidotu treniņu. Gadījumā, kad lietotājam ir izveidoti treniņi, sistēma izvadā tos viena sarakstā.

Kad lietotājs uzspiež uz dzēšanas pogu, uz ekrāna tiek izvadīts paziņojums par to, ka ja lietotājs apstiprina treniņa dzēšanu, tad tā tiek dzēsta no sistēmas. Gadījumā, kad lietotājs neapstiprina treniņa dzēšanu, operācija tiek pārtraukta.

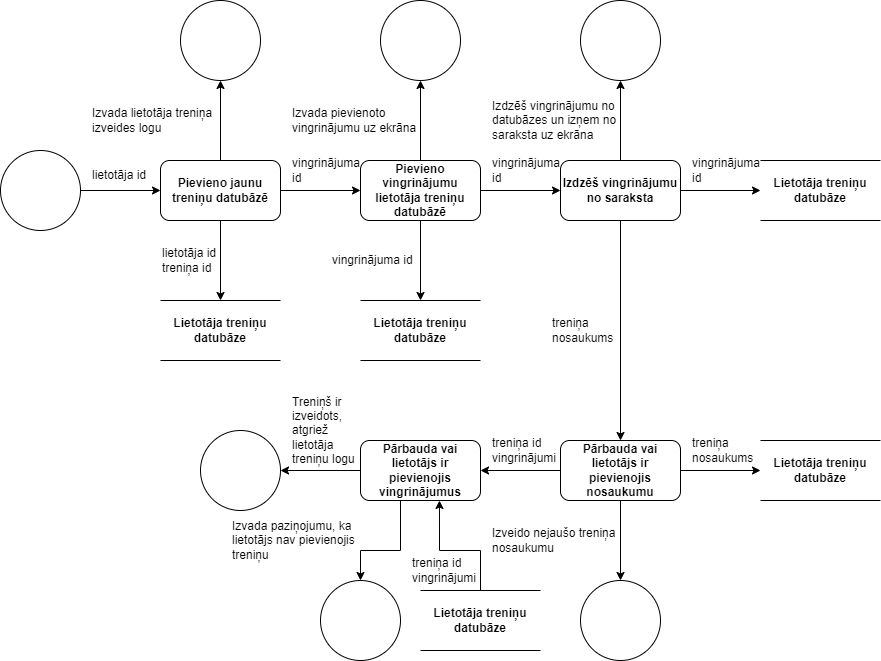
****4.13. Atsevišķa lietotāja treniņa dzēšanas datu plūsmu diagramma

**11. Lietotāja treniņa izveidošana**

Lietotājam jābūt iespējai izveidot treniņu. Lietotājam būs jāsastāda vingrinājuma sarakstu no piedāvātiem vingrinājumiem un jāuzraksta nosaukumu. Lietotāja treniņa izveidošanas process notiks sekojoši (skat. 4.14. att.).

Pirmkārt, treniņš tiek pievienots datu bāzē. Kad lietotājs izvēlas vingrinājumu no saraksta, tas tiek pievienots datu bāze un izvadīts uz ekrāna. Ja vingrinājums tiek izdzēsts, tad tas tiek izņemts no datu bāzes un noņemts no ekrāna.

Kad lietotājs apstiprina, ka treniņš ir izveidots, sistēma pārbauda, vai tas ir pievienojis savu nosaukumu. Gadījumā, kad lietotājs nav ievadījis nosaukumu, tas tiek izveidots nejauši un abos gadījumos, nosaukums tiek pievienots datu bāzē. Pēc tam tiek pārbaudīts vai lietotājs ir pievienojis vingrinājumus, ja tas netīk izdarīts, tad uz ekrāna tiek izvadīts paziņojums par to, ka lietotājs nav pievienojis vingrinājumus, ja tie bija pievienoti, tad treniņa izveidošanas process ir pabeigts un uz ekrāna tiek atgriezts lietotāja treniņu saraksts.

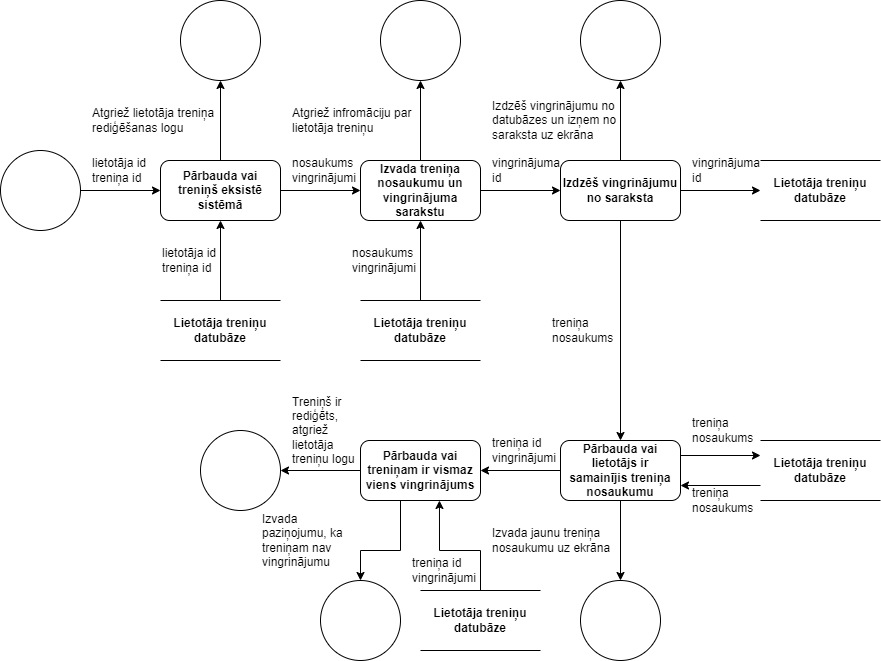
4.14. att. Lietotāja treniņa izveidošanas datu plūsmu diagramma

**12. Lietotāja treniņa rediģēšana**

Lietotājam jābūt iespējai rediģēt savus treniņus. Rediģēšanas process notiks līdzīgi izveides procesam. Lietotāja treniņa rediģēšanas process notiks sekojoši (skat. 4.15. att.).

Sākumā, lietotne pārbauda vai izvēlētais treniņš rediģēšanai eksistē sistēmā. Ja treniņš eksistē sistēmā, tad tiek atgriezts treniņa rediģēšanas logs, treniņa nosaukums un vingrinājumu saraksts. Ja vingrinājums tiek izdzēsts, tad tas tiek izņemts no datu bāzes un noņemts no ekrāna.

Lietotne pārbauda vai lietotājs ir samainījis treniņa nosaukumu. Gadījumā, kad tas ir noticis, tad jauns nosaukums tiek pievienots datu bāzē. Pēc tam tiek pārbaudīts vai treniņam ir vismaz viens vingrinājums, ja tas ir patiess, tad uz ekrāna tiek izvadīts paziņojums par to, ka treniņam nav pievienoto vingrinājumu, ja treniņam ir vismaz viens vingrinājums, tad treniņa rediģēšanas process ir pabeigts un uz ekrāna tiek atgriezts lietotāja treniņu saraksts.

4.15. att. Lietotāja treniņa rediģēšanas datu plūsmu diagramma

**13. Standarta (jau izveidota) treniņu attēlošana**

Lietotājam jābūt iespējai apskatīt standartus (jau izveidotos) treniņus. Visi standarti treniņi būs izvadīti viena sarakstā. Standartu treniņu saraksta apskatīšana notiks sekojoši (skat. 4.16. att.).

Pirmkārt, sistēma pārbauda vai datu bāzē ir standarti treniņi. Ja tādu nav, tad uz ekrāna izvada kļūdas paziņojumu. Kad datu bāzē ir standarti treniņi, sistēma izvadā tos viena sarakstā, parādot tā nosaukumu, vingrinājuma skaitu un aprakstu. Lietotāja treniņi ir izvadīti.

4.16. att. Standarta treniņu attēlošanas datu plūsmu diagramma

**14. Treniņa izpilde**

Lietotājam jābūt iespējai pildīt treniņus. Treniņa izpildes laikā, lietotājam būs iespēja pievienot un dzēst piegājienus, ievadīt atkārtojuma skaitu un inventāra svaru, ka arī apskatīt vingrinājuma izpildes aprakstu. Visi standarti treniņi būs izvadīti viena sarakstā. Treniņu izpildes process notiks sekojoši (skat. 4.17. att. un 4.18. att.).

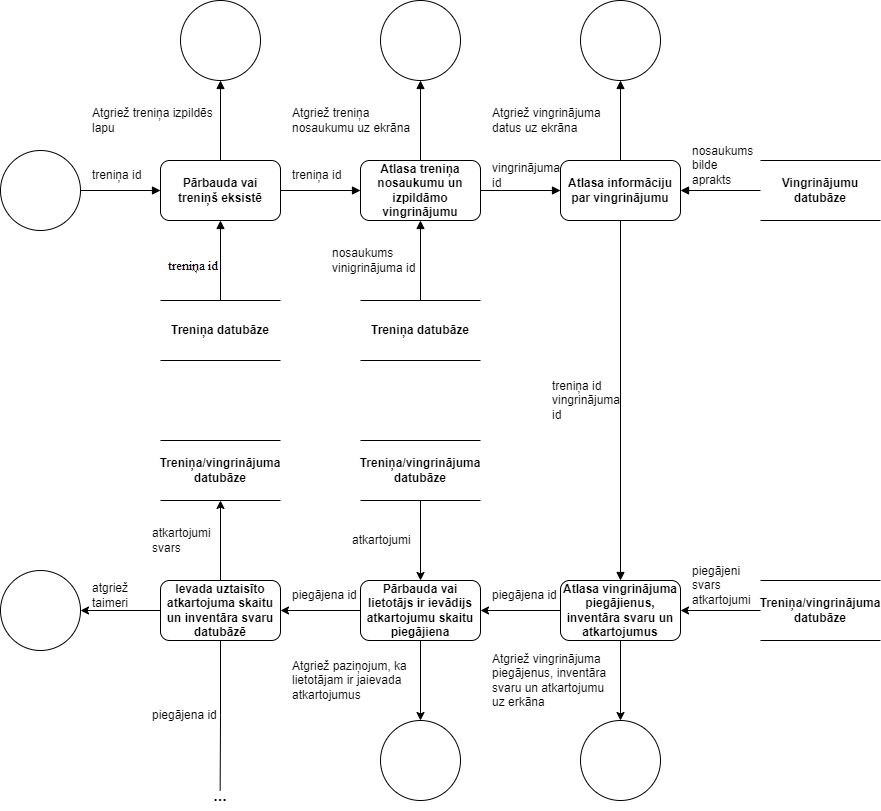
Pirmkārt, lietotne pārbauda vai treniņš eksistē sistēmā. Pārbaude ir taisīta tikai lai sistēmai būtu iespēja tikt pie treniņa un vingrinājuma datiem. Lietotne atlasa treniņa nosaukumu un izpildāmo vingrinājumā unikālo atslēgu (ID) un atgriež tos uz ekrāna.

Nākamais solis ir informācijas par izpildāmo vingrinājumu atlase. Kad vingrinājuma nosaukums, bilde un apraksts ir atlasīti, tie tiek izvadīti uz ekrāna. Pēc tam tiek atlasīti vingrinājuma piegājienu skaits, inventāra svars un atkārtojumu daudzums un izvadīti uz ekrāna.

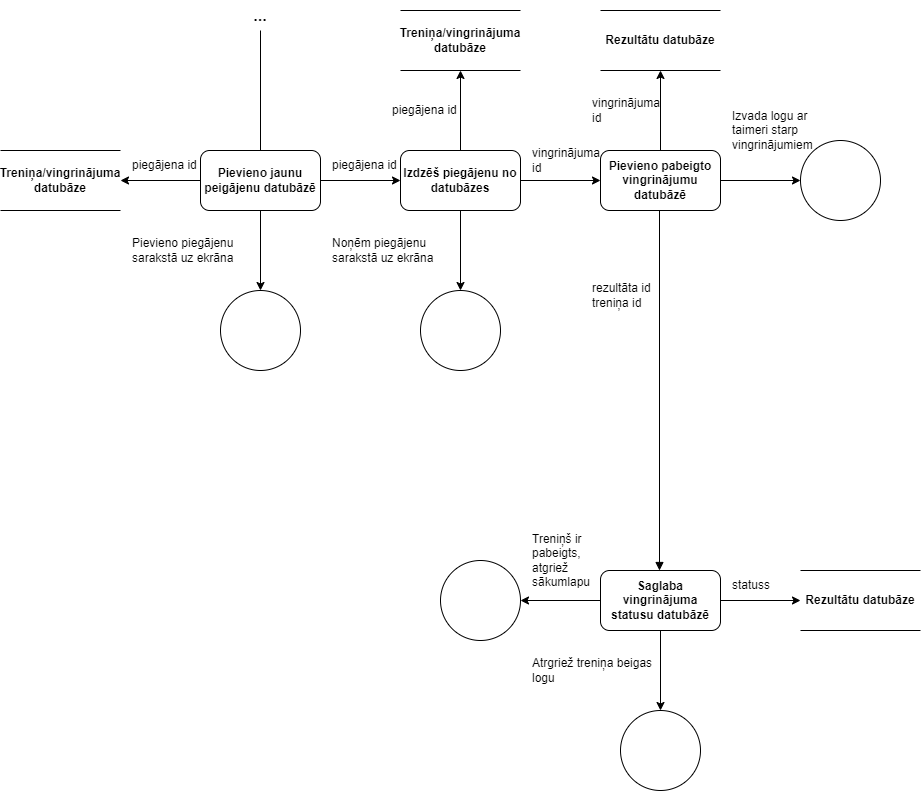
Kad lietotājs pabeidz piegājienu, lietotne pārbauda, vai lietotājs ir ievadījis atkārtojumā daudzumu, ja tie ne tiek ievadīti vai ir vienādi ar nulli, tad sistēma paziņo lietotājam, ka viņam ir jāpieraksta tos. Gadījumā, kad tie ir ievadīti, lietotne saglāba tos datu bāzē atkārtojumus un inventāra svaru datu bāzē un atgriež starp piegājienu taimeri.

Kad lietotājs pievieno jaunu piegājienu, tas tiek ierakstīts datu bāzē un pievienots piegājienu sarakstā uz ekrāna. Gadījumā, kad lietotājs izdzēš to, tas tiek izdzēsts no datu bāzes un no piegājienu sarakstā uz ekrāna.

Kad visi piegājieni ir pabeigti, lietotne pievieno vingrinājumu datu bāzē pie pabeigtiem treniņiem un atgriež starp logu, kurā var redzēt taimeri un nākamo vingrinājumu.

****Kad visi vingrinājumi ir pabeigti, tiek izvadīts nobeiguma logs un treniņa rezultātos tiek pievienots status (*“finished (pabeigts)” –* kad lietotājs nokļuva līdz nobeiguma logam, *“unfinished (nepabeigts)”* – kad lietotājs nenokļuva līdz nobeiguma logam un treniņa process tika pārtraukts un *“cancelled”* – kad lietotājs pārtrauca treniņa procesu). Treniņš ir pabeigts, atgriež lietotāju galveno logu.

4.17. Treniņa izpildes procesa datu plūsmu diagramma (1. daļa)

****

4.18. Treniņa izpildes procesa datu plūsmu diagramma (2. daļa)

**5. DATU STRUKTŪRU APRAKSTS**

Datu bāze bija taisīta izmantojot Google Firebase servisa Realtime Database rīku, kas saglāba datu bāzi JSON failā, NoSQL valodā. NoSQL datu bāze tabulas vietā izmanto klases un laukiem nav izmēra. Realtime Database izmanto sekojošus datu tipus: STRING, OBJECT, NUMBER, BOOLEAN un AUTO.

Datu bāze sastāv no 8 klasēm. Tā satur sevī informāciju par lietotājiem, standartiem un lietotāja izveidotajiem treniņiem, lietotāja rezultātiem un vingrinājumiem. Datu bāzes klases shēma ir dota 1. pielikumā

Klase “users” sevī ietver visus lietotnes nepieciešamos lietotāja datus.

5.1. klase

Klases **“users”** struktūra

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Lauka nosaukums** | **Tips** | **Apraksts** |
| 1 | id | AUTO | Reģistrētā lietotāja unikāla atslēga |
| 2 | email | STRING | Reģistrētā lietotāja e-pasts |
| 3 | name | STRING | Reģistrētā lietotāja vārds |
| 4 | weight | NUMBER | Reģistrētā lietotāja svars |
| 5 | height | NUMBER | Reģistrētā lietotāja augums |

Klase “[usersWorkouts](https://console.firebase.google.com/u/1/project/fitnessapp-diploms/database/fitnessapp-diploms-default-rtdb/data/~2FusersWorkouts)” satur datus par treniņiem, kurus izveidoja lietotājs.

5.2. klase

Klases **“usersWorkouts”** struktūra

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Lauka nosaukums** | **Tips** | **Apraksts** |
| 1 | id | AUTO | Lietotāja treniņa unikāla atslēga |
| 2 | user | AUTO | Reģistrētā lietotāja unikāla atslēga |
| 3 | name | STRING | Lietotāja treniņa nosaukums |

Klase “[premadeWorkouts](https://console.firebase.google.com/u/1/project/fitnessapp-diploms/database/fitnessapp-diploms-default-rtdb/data/~2FpremadeWorkouts)” satur datus par standartiem treniņiem. Lauka “image” glābās saite uz bildi, kas glābās Firebase Storage.

5.3. klase

Klases **“premadeWorkouts”** struktūra

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Lauka nosaukums** | **Tips** | **Apraksts** |
| 1 | id | AUTO | Standarta treniņa unikāla atslēga |
| 2 | name | STRING | Standarta treniņa nosaukums |
| 3 | muscles | STRING | Standarta treniņa akcentēti muskuļi |
| 4 | description | STRING | Standarta treniņa apraksts |
| 5 | image | STRING | Saite uz standarta treniņa ikonu |

Klase “resultsExercises” satur datus par vingrinājumiem, kas bija izpildīti treniņa laikā.

5.4. klase

Klases **“resultsExercises”** struktūra

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Lauka nosaukums** | **Tips** | **Apraksts** |
| 1 | id | AUTO | Rezultāta un vingrinājuma unikāla atslēga |
| 2 | results | AUTO | Rezultāta unikāla atslēga |
| 3 | exercises | AUTO | Vingrinājuma unikāla atslēga |

Klase “results” satur datus par lietotāja pabeigta treniņa rezultātiem.

5.5. klase

Klases **“results”** struktūra

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Lauka nosaukums** | **Tips** | **Apraksts** |
| 1 | id | AUTO | Rezultāta unikāla atslēga |
| 2 | user | AUTO | Reģistrētā lietotāja unikāla atslēga |
| 3 | workout | AUTO | Treniņa unikāla atslēga |
| 4 | date | STRING | Datums, kad treniņš bija pabeigts |

Klase “exercises” satur datus par vingrinājumiem. Laukā “image” glābās saite uz bildi, kas glābās Firebase Storage.

5.6. klase

Klases **“exercises”** struktūra

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Lauka nosaukums** | **Tips** | **Apraksts** |
| 1 | id | AUTO | Vingrinājuma unikāla atslēga |
| 2 | name | STRING | Vingrinājuma nosaukums |
| 3 | muscles | STRING | Vingrinājuma akcentēti muskuļi |
| 4 | equipment | STRING | Vingrinājuma izmantots inventārs |
| 5 | description | STRING | Vingrinājuma apraksts |
| 6 | image | STRING | Saite uz vingrinājuma bildi |

Klase “[workoutPremadeExercises](https://console.firebase.google.com/u/1/project/fitnessapp-diploms/database/fitnessapp-diploms-default-rtdb/data/~2FusersWorkouts)” tika veidota starp klasēm “premadeWorkouts” un “exercises”, jo starp tam klasēm ir relācija daudzi-pret-daudziem. Klase satur datus par standarta treniņu izpildēs procesu.

5.7. klase

Klases **“workoutPremadeExercises”** struktūra

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Lauka nosaukums** | **Tips** | **Apraksts** |
| 1 | workout | AUTO | Treniņa unikāla atslēga |
| 2 | exercises | AUTO | Vingrinājuma unikāla atslēga |
| 3 | sets | NUMBER | Piegājienu skaits |
| 4 | reps | NUMBER | Atkārtojumu skaits |
| 5 | weight | NUMBER | Svars ar kuru bija pildīts vingrinājums |

Klase “[workoutUsersExercises](https://console.firebase.google.com/u/1/project/fitnessapp-diploms/database/fitnessapp-diploms-default-rtdb/data/~2FusersWorkouts)” tika veidota starp klasēm “usersWorkouts” un “exercises”, jo starp tam klasēm ir relācija daudzi-pret-daudziem. Klase satur datus par lietotāja treniņu izpildēs procesu.

5.8. klase

Klases **“workoutUsersExercises”** struktūra

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr.** | **Lauka nosaukums** | **Tips** | **Apraksts** |
| 1 | workout | AUTO | Treniņa unikāla atslēga |
| 2 | user | AUTO | Reģistrētā lietotāja unikāla atslēga |
| 3 | exercises | AUTO | Vingrinājuma unikāla atslēga |
| 4 | sets | NUMBER | Piegājienu skaits |
| 5 | reps | NUMBER | Atkārtojumu skaits |
| 6 | weight | NUMBER | Svars ar kuru bija pildīts vingrinājums |

**6. LIETOTĀJA CEĻVEDIS**

**6.1. Sistēmas prasības aparatūrai un programmatūrai**

Lietotnei ir vienkāršs un viegls grafiskais lietotāja interfeiss (GUI) un dizains, tāpēc tā ir izstrādāta uz Android 5.0 Lollipop. 98.6% Android ierīču lietotājiem ir Android 5.0 vai jaunākas versijas. Android 5.0 atbalsta lielāku daļu no jaunām tehnoloģijām, kas ir nepieciešami vienkāršās Android lietotnes izstrādei. Kā var saprast, lietotājam, lai izmantotu šo lietotni, ir vajadzīga mobila ierīce, kas atbalsta Android 5.0 Lollipop vai jaunāks.

6.1. tabula

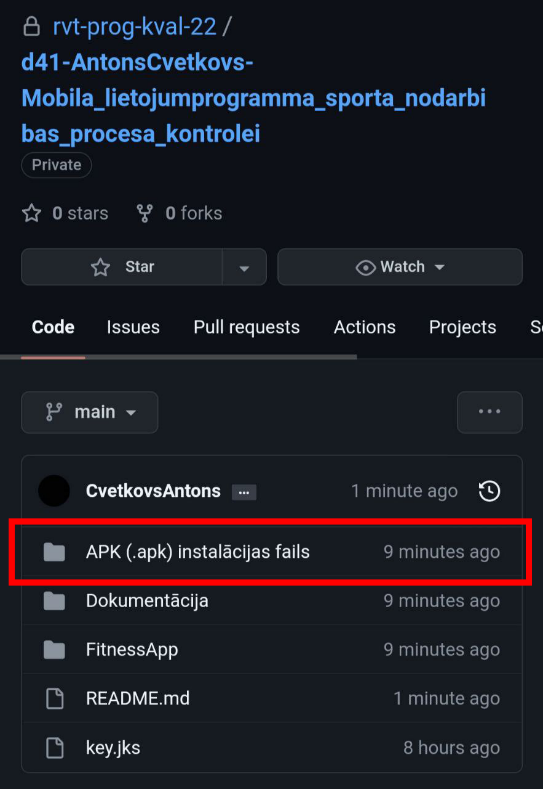
Lietotnes atbalstītās uz dažādām operētājsistēmām

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Android** | **iOS** | **Mac OS X** | **Windows** | **Linux** |
| Atbalstīts | Nav atbalstīts | Nav atbalstīts | Nav atbalstīts | Nav atbalstīts |

**6.2. Sistēmas instalācija un palaišana**

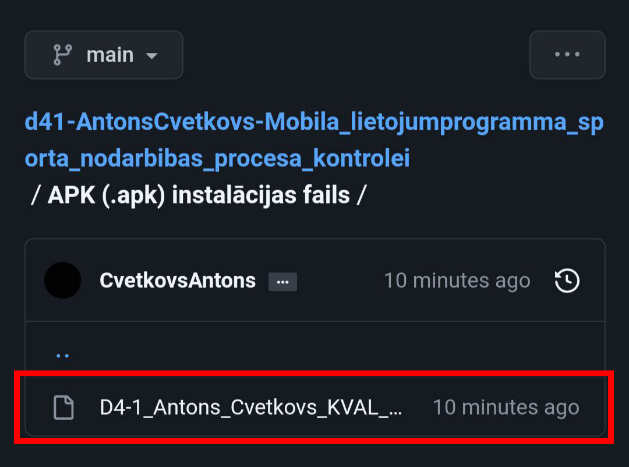
Lai uzinstalētu lietotni ir nepieciešam lejupielādēt APK (.apk) instalācijas failu no kvalifikācijas darba GitHub repozitorijos un palaist to.

**Instalācijas instrukcija**

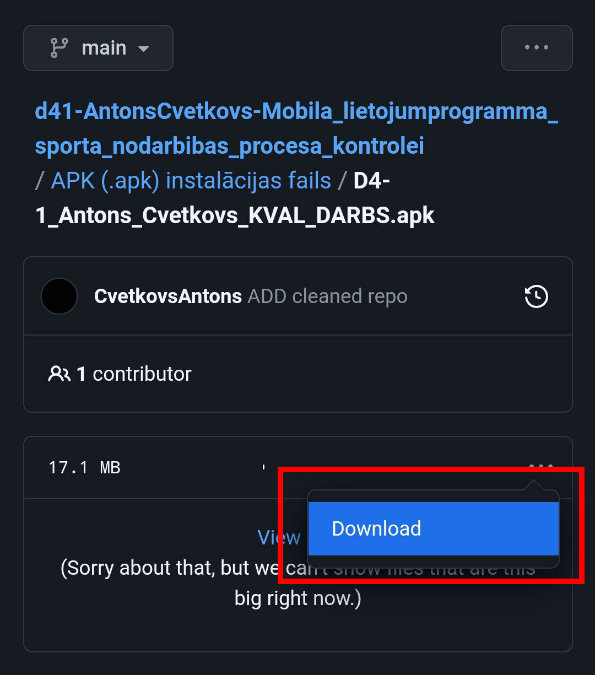
1. Kvalifikācijas darba repozitorijā atvērt APK (.apk) instalācijas fails mapi

6.1. att. APK (.apk) instalācijas fails mape

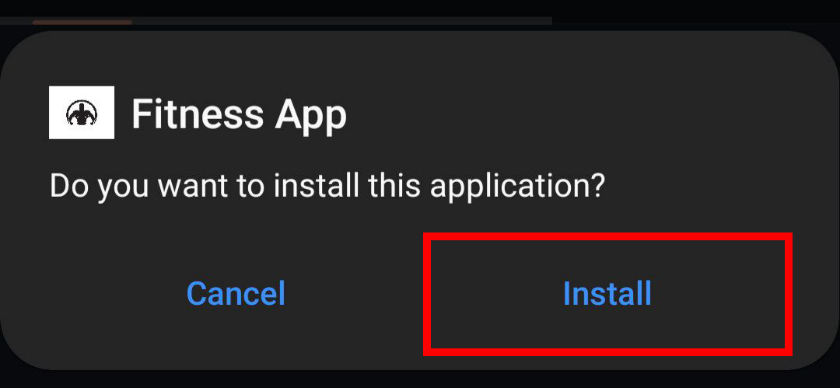
1. Atvērt failu *D4-1\_Antons\_Cvetkovs\_KVAL\_DARBS.apk*

**

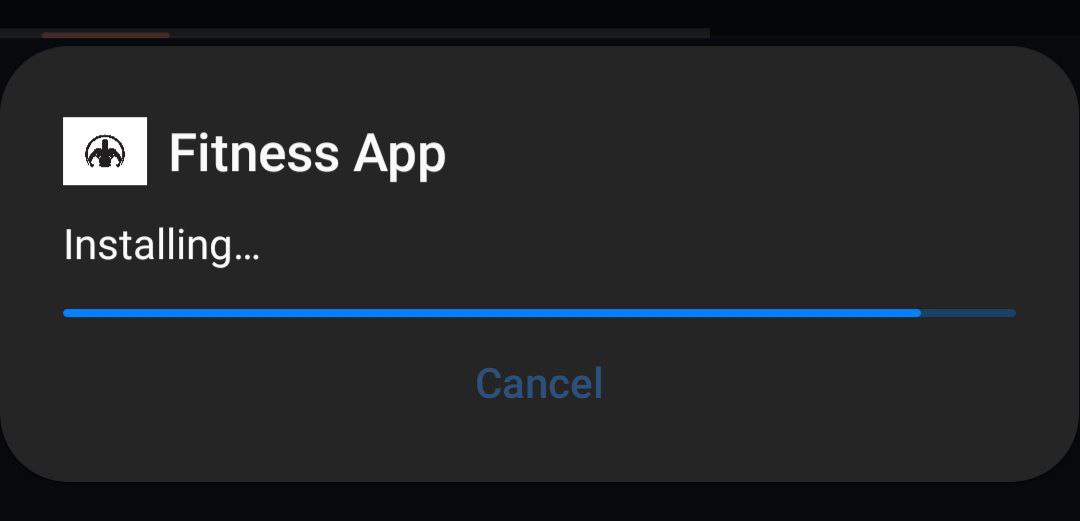
6.2. att. D4-1\_Antons\_Cvetkovs\_KVAL\_DARBS.apk

1. Lejupielādējiet failu

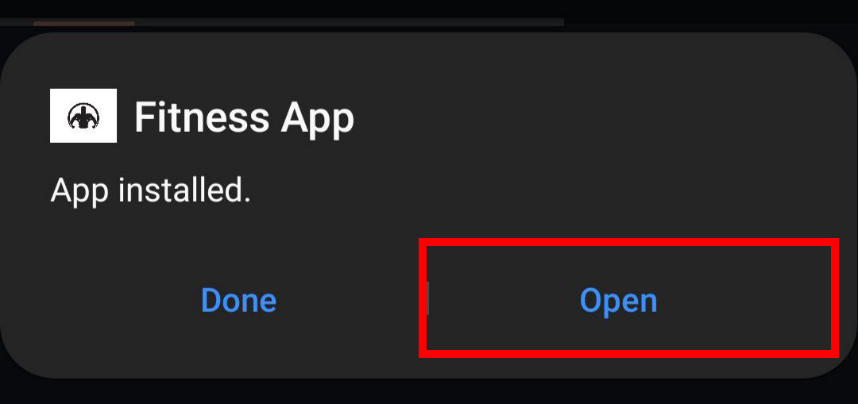
6.3. att. Instalācijas faila lejupielādēšana

1. Atvērāt instalācijas failu un nospiedāt uz *Install*, lai uzsāktu lietotnes instalāciju un uzgaidiet tā nobeigumu

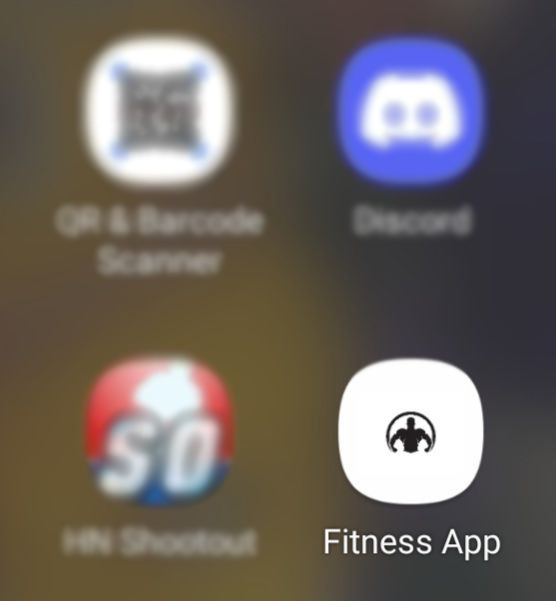
6.4. att. Instalācijas sākums



6.5. att. Lietotnes instalācija

1. Kad instalācija ir pabeigta atveriet lietotni, nospiežot uz *Open* vai caur lietotnes ikonu jūsu ierīces izvēlnē

6.6. att. Lietotnes palaišana uzreiz pēc instalācijas

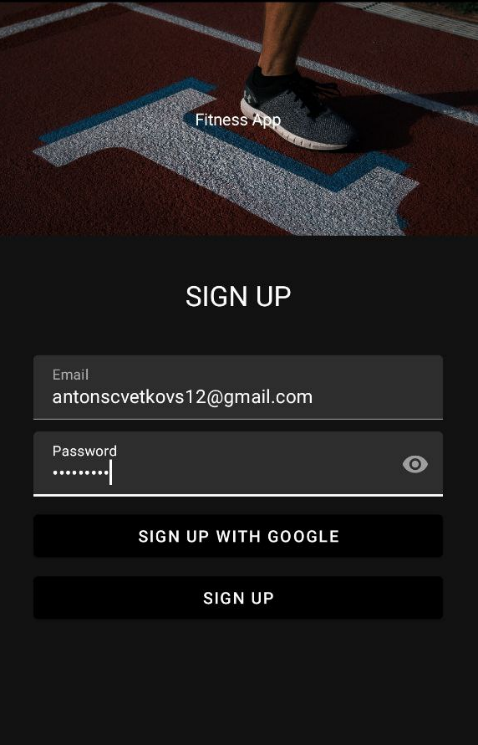


6.7. att. Lietotnes palaišana caur ikonu

**6.3. Programmas apraksts**

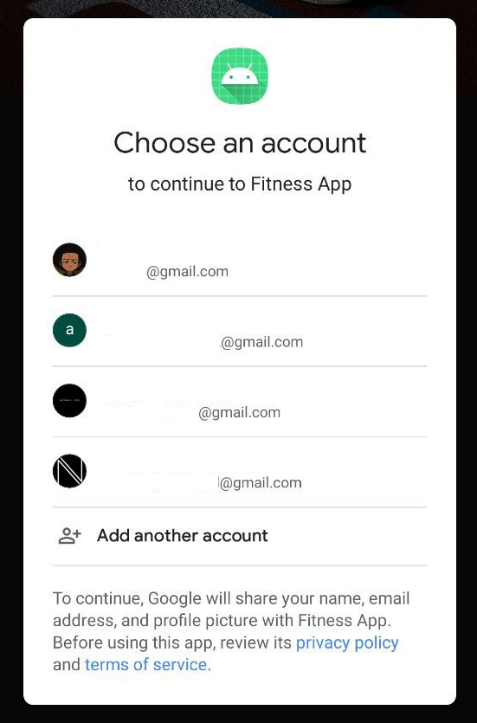
**Reģistrācijas/pieslēgšanās logs**

Kad lietotājs palaiž lietotni, uz ekrāna tiek izvadīts reģistrācija/pieslēgšanās logs (skat. 6.8. att.). Logā var atrast e-pastas un paroles ievades laukus, reģistrācijas/pieslēgšanas caur Google un reģistrācijas apstiprinājuma pogas..

****

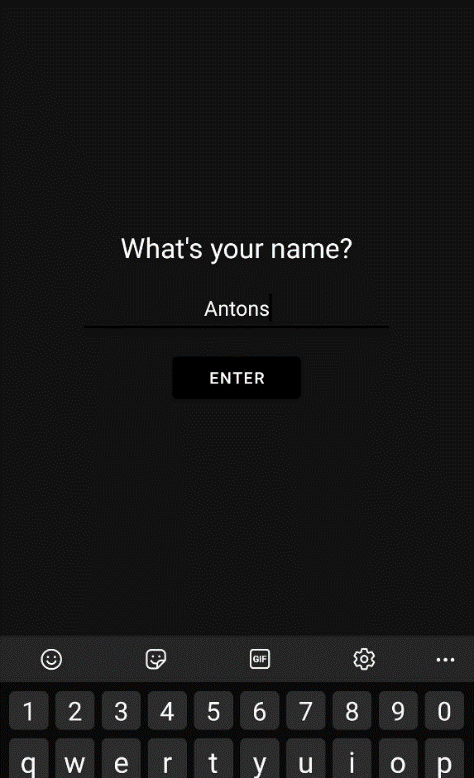
6.8. att. Reģistrācijas/pieslēgšanas logs

Kad lietotājs uzspied uz reģistrācija caur Google (Sign up with Google) pogu, uz ekrāna izvadās logs ar lietotāja Google profiliem (skat. 6.9. att.).

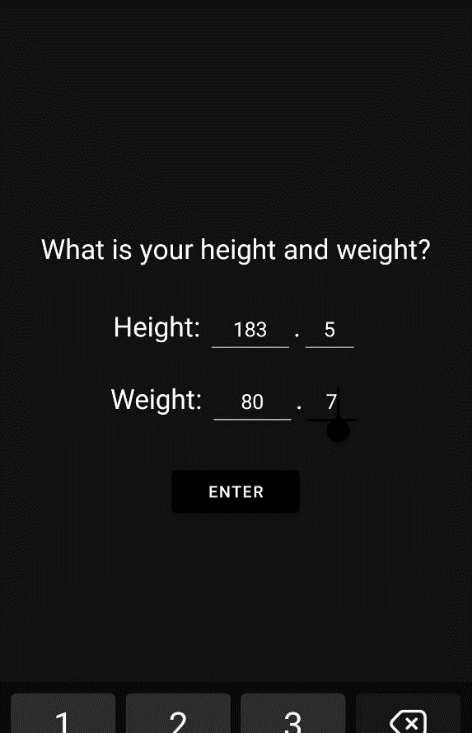


6.9. att. Reģistrācija/pieslēgšana caur Google

**Lietotāja datu ievade**

Pēc sekmīgas reģistrācijas, uz ekrāna izvadā lietotāja vārda ievades logu (skat. 6.10. att.). Kurā var atrast vārda ievades lauku. Kad lietotājs sekmīgi ievada savu vārdu uz ekrānu izvadā viņa svara un auguma ievades logu (skat. 6.11. att.).

6.10. att. Lietotāja vārda ievade



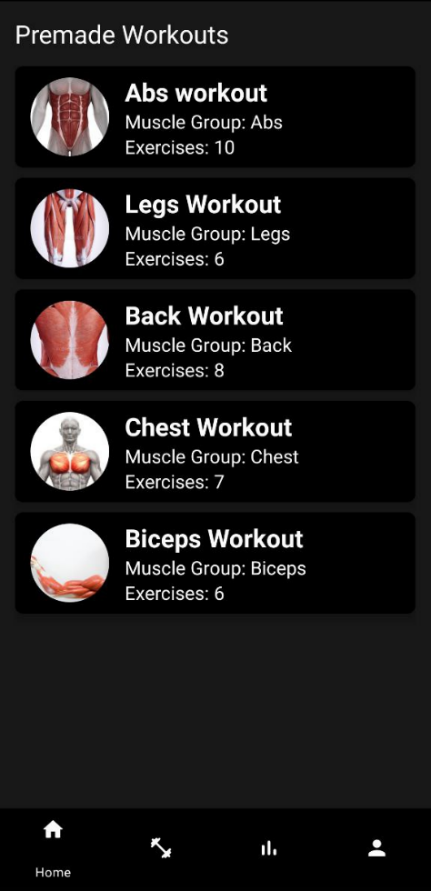
6.11. att. Lietotāja auguma un svara ievades logs

**Apakšizvēlne**

****Lai pārvietoties pa lietotni tika izveidota apakšizvēlne (skat. 6.12. att.) ar četrām sekcijām – *Home (galvenais logs)* (skat. 6.13. att.), *Your Workouts (lietotāja treniņu logs)* (skat. 6.14. att. un 6.15. att.), *Results (lietotāja rezultātu logs)* (skat. 6.17. att. un 6.18. att.) un *Profile (lietotāja profila logs)* (skat. 6.20. att.).

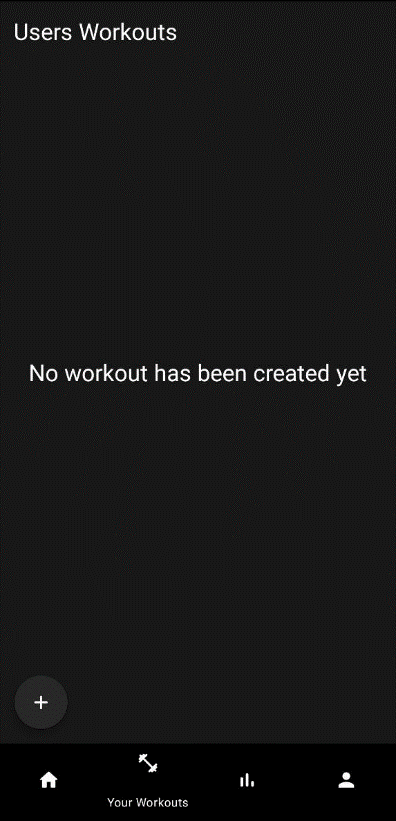
6.12. att. Apakšizvēlne

**Galvenais logs**

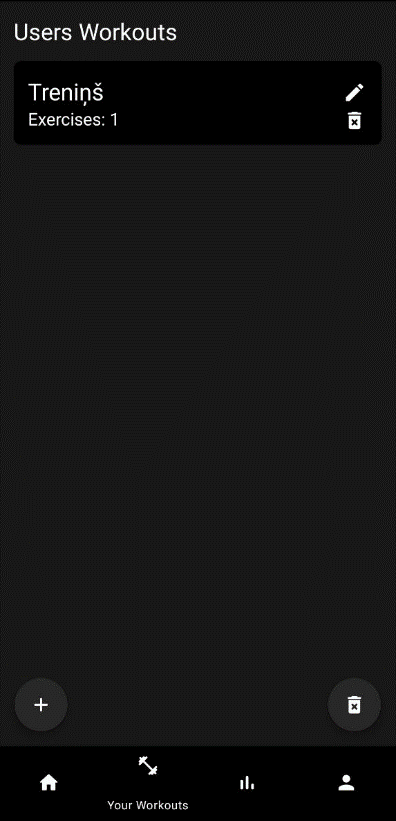
Kad lietotājs sekmīgi ievada savus datus, uz ekrānu tiek izvadīts galvenais logs (skat. 6.13. att.) ar standartu, iepriekš izveidotu treniņu sarakstu. Sarakstā var redzēt treniņa nosaukumu, muskuļu grupu, kas ir trenēta un vingrinājumu skaitu. Apakšā atrodas izvēlne. Kad lietotājs nospiež uz vienu no treniņiem uz ekrānu izvadīs treniņa priekšskatījuma logu.

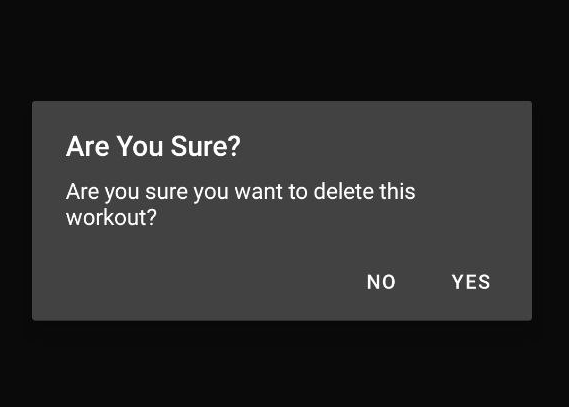
6.13. att. Galvenais logs

**Lietotāja treniņu logs**

****Kad lietotājs izvēlas *Your Workouts* sekciju, apakšizvēlnē, uz ekrāna tiek izvadīts logs ar lietotāja treniņu sarakstu. Kad lietotājam nav izveidotu treniņu uz ekrāna tiek izvadīts teksts “No workout has been created yet” (Nav izveidotu treniņu) (skat. 6.14. att.) un pogu, apakšējā kreisajā stūri, uzspiežot uz kuru, sāksies treniņu izveides process. Gadījumā, kad lietotājam ir izveidoti treniņi, tiek izvadīts saraksts ar lietotāja treniņiem, kurā būs pieejama informācija par tiem – nosaukums un vingrinājumu daudzums (skat. 6.15. att.). Ka arī rediģēšanas un viena treniņa dzēšanas pogas. Apakšēja labajā stūrī tiek izvadīta poga, uzspiežot uz kuru, būs izdzēsti visi lietotāja treniņi.

6.14. att. Lietotāja treniņu tukšs logs

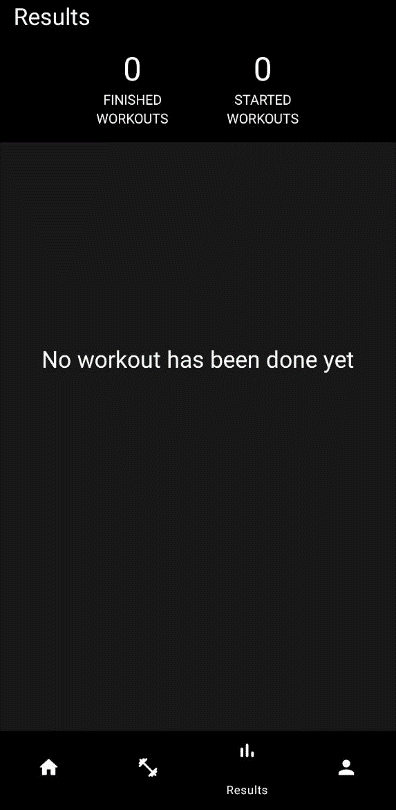


****6.15. att. Lietotāja treniņu aizpildīts logs

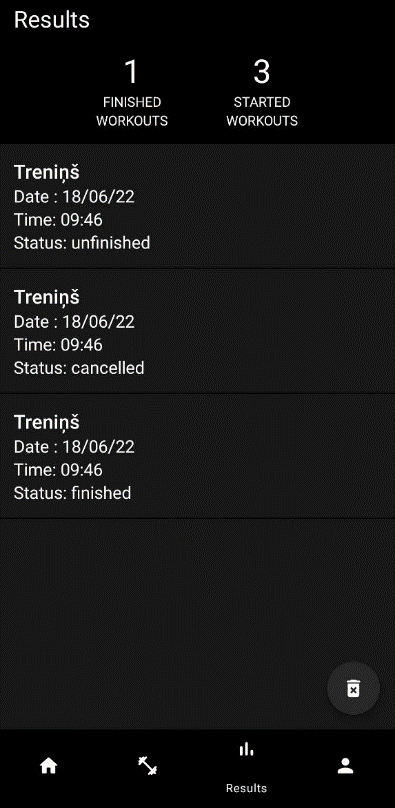
6.16. att. Viena lietotāja treniņa dzēšanas apstiprinājuma logs

**Lietotāja rezultātu logs**

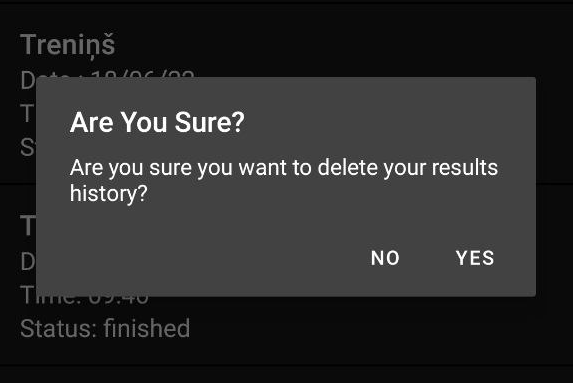
Kad lietotājs izvēlas *Results* sekciju, apakšizvēlnē, uz ekrāna tiek izvadīts logs ar lietotāja rezultātu sarakstu. Kad lietotājam nav rezultātu uz ekrāna tiek izvadīts teksts “No workout has been done yet” (Nav pabeigto treniņu) (skat. 6.17. att.). Paša augšā var redzēt lietotāja pabeigto un uzsāktu treniņa skaitu. Gadījumā, kad lietotājam ir rezultāti, tiek izvadīts saraksts ar tā rezultātiem, kurā būs pieejama informācija par tiem – treniņa nosaukums, izpildes datums, sākuma laiks un statuss (skat. 6.18. att.). Ka arī, apakšēja labajā stūrī tiek izvadīta poga, uzspiežot uz kuru, būs izdzēsti visi lietotāja rezultāti.

****

6.17. att. Lietotāja rezultātu tukšs logs

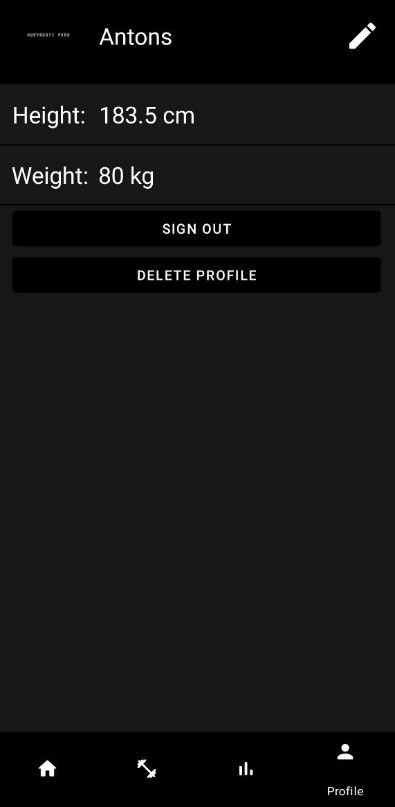
****

6.18. Lietotāja rezultātu aizpildīts logs

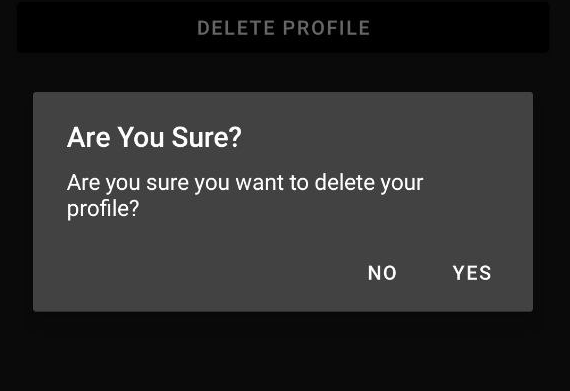
****

6.19. att. Lietotāja rezultātu dzēšanas apstiprinājuma logs

**Lietotāja profila logs**

****Kad lietotājs izvēlas *Profile* sekciju, apakšizvēlnē, uz ekrāna izvadās logs ar informāciju par lietotāju (skat. 6.20. att.) – vārds, augums, svars un Google konta bilde. Zem svara un auguma atrodas izrakstīšanas (sesijas pārtraukšanas) un profila dzēšanas pogas. Labajā augšējā stūrī atrodas poga, kas atver lietotāja datu rediģēšanas logu (skat. 6.22. att.), tas tiek aprakstīts tālāk.

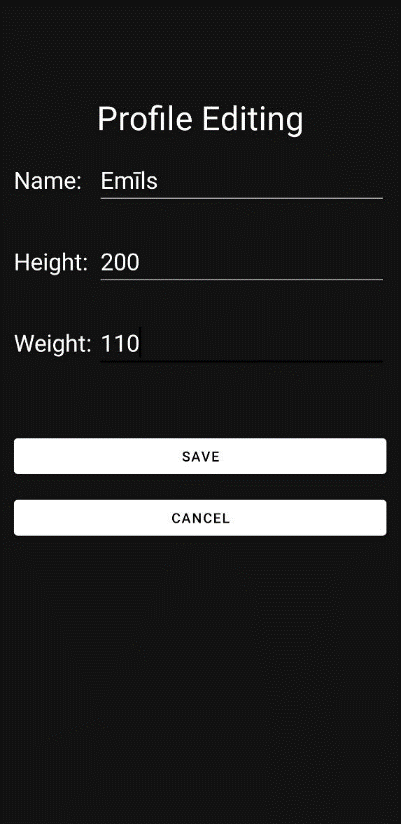
6.20. att. Lietotāja profila logs



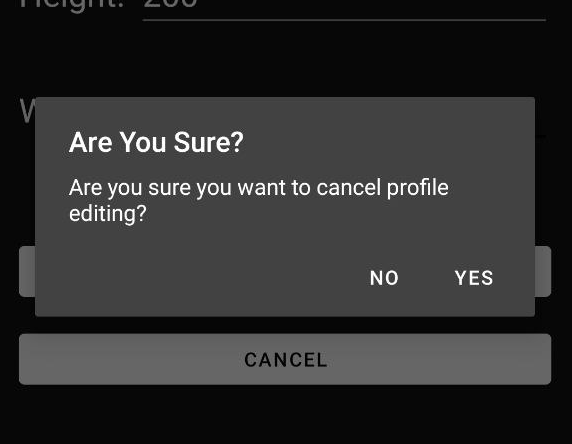
6.21. att. Lietotāja profila dzēšanas apstiprinājuma logs

**Lietotāja profila rediģēšanas logs**

Kad lietotāja profila rediģēšanas logs (skat. 6.22. att.) tiek izvadīts, uz ekrāna paradās trīs ievades lauki – lietotāja vārda, auguma un svara. Katra laukā tiek izvadīti sekojoši tiem dati. Zem ievades laukiem atrodas izmaiņu saglabāšanas un atcelšanas pogas.

****

6.22. att. Lietotāja profila rediģēšanas logs (ar rediģētiem datiem)

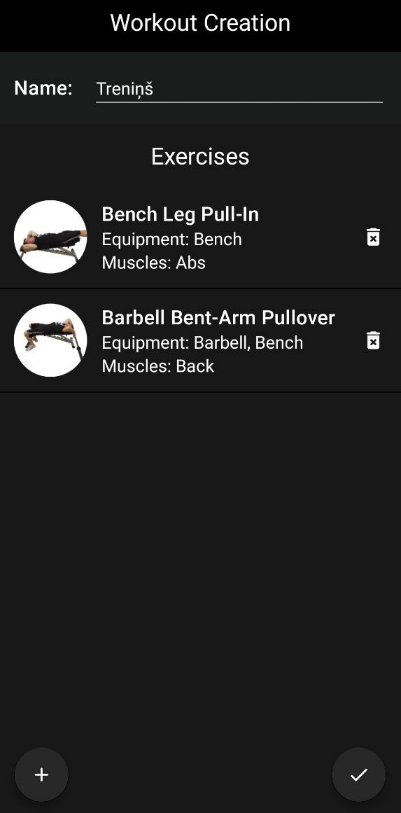
****

6.23. att. Lietotāja profila rediģēšanas atcelšanas apstiprinājuma logs

**Treniņa izveides logs**

****Treniņa izveides logā tiek izvadīti treniņa nosaukuma ievades logs, izvēlēto vingrinājumu saraksts, ar informāciju par tiem – bilde, nosaukums, inventārs un trenēta muskuļu grupa. Pie katra vingrinājuma ir tā dzēšanas poga, uzspiežot uz kuru, vingrinājums tiek izņemts no saraksta. Kreisajā-apakšējā stūrī atrodas vingrinājuma pievienošanas poga un labajā-apakšējā stūrī treniņa izveides beigšanas poga (skat. 6.24. att. un 6.25. att.).

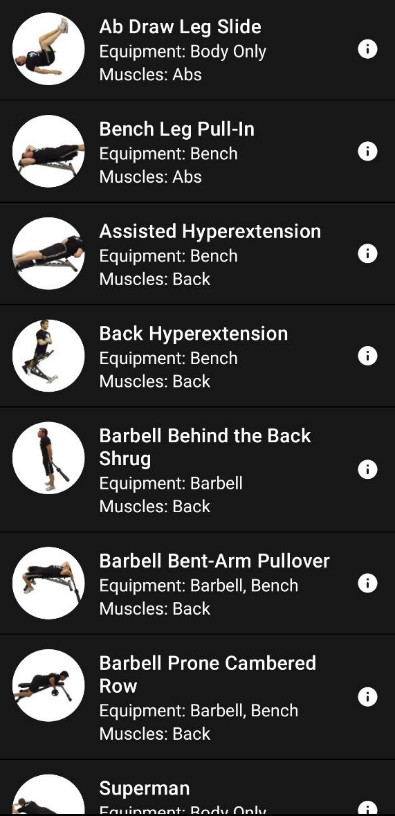
6.24. att. Treniņa izveides tukšs logs

****

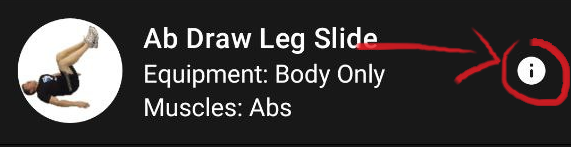
6.25. att. Treniņa izveides aizpildīts logs

**Vingrinājuma izvēlēs logs**

Kad lietotājs nospiež uz vingrinājuma pievienošanas pogu treniņa izveides logā, uz ekrāna tiek izvadīts vingrinājuma izvēlēs logs (skat. 6.26. att.). Par katru vingrinājumu tiek izvadīti tā nosaukums, bilde, inventārs un trenēta muskuļu grupa. Uzspiežot uz vingrinājumu, tas tiek pievienots treniņa vingrinājuma sarakstā. Lai apskatītu visu informāciju par vingrinājumu, lietotājam ir jāuzspiež uz sekojošu pogu (skat. 6.27. att.).

****

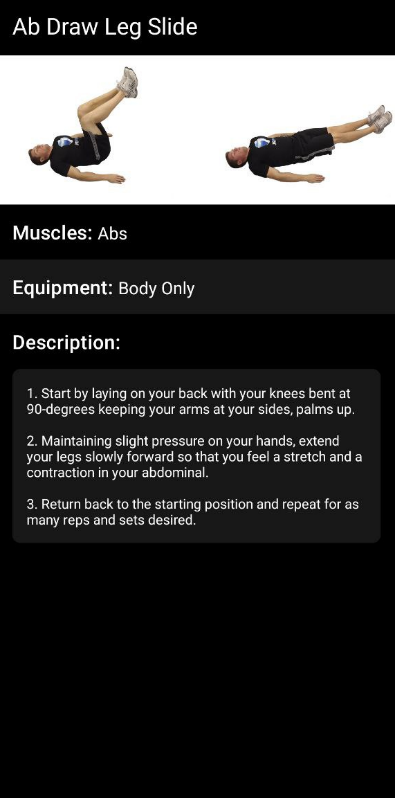
6.26. att. Vingrinājuma izvēlēs logs

****

6.27. att. Vingrinājuma apraksta poga

**Vingrinājuma apraksta logs**

Lietotājam jābūt iespējai apskatīt informāciju par vingrinājumiem. Tā priekš ir izveidots vingrinājuma apraksta logs (skat. 6.28. att.), kurā tiek izvadīti vingrinājuma apraksts, bilde ar vingrinājuma pareizas izpildes tehnikas piemērs, trenēto muskuļu grupa, nepieciešams inventārs un vingrinājuma izpildes aprakstu.

****

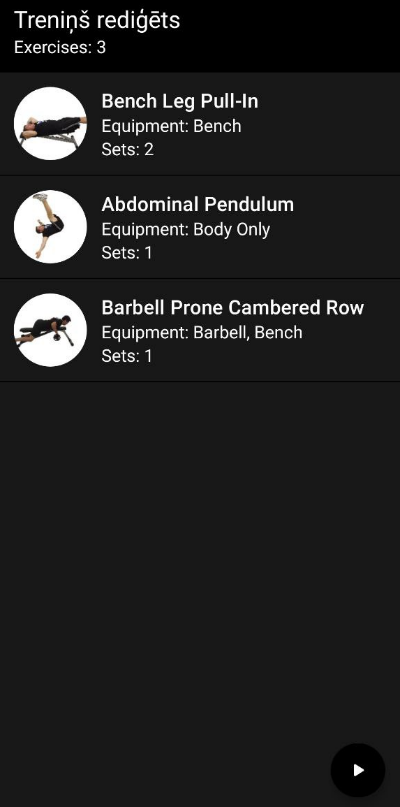
6.28. att. Vingrinājuma apraksta logs

**Treniņa rediģēšanas logs**

Treniņa rediģēšanas logs izskatās un strādā līdzīgi treniņa izveides (skat. 6.24. att. un 6.25. att.) logam. Lietotāja var nomainīt treniņa nosaukumu, pievienot un dzēst vingrinājumus. Vienīga atšķirība ir loga nosaukumā.

**Treniņa priekšskatījuma logs**

Pirms katra treniņa, uz ekrāna tiek izvadīts tā priekšskatījums (skat. 6.29. att.) ar treniņa nosaukumu, vingrinājuma skaitu un sarakstu. Labajā-apakšējā stūrī var atrast treniņa sākuma pogu, uzspiežot uz kuru, tiek uzsākts treniņa process.

****

6.29. att. Treniņa priekšskatījuma logs

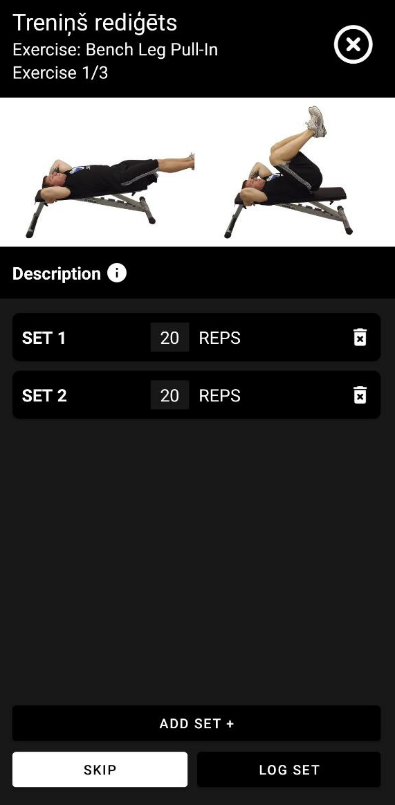
**Treniņa izpildes procesa apraksts**

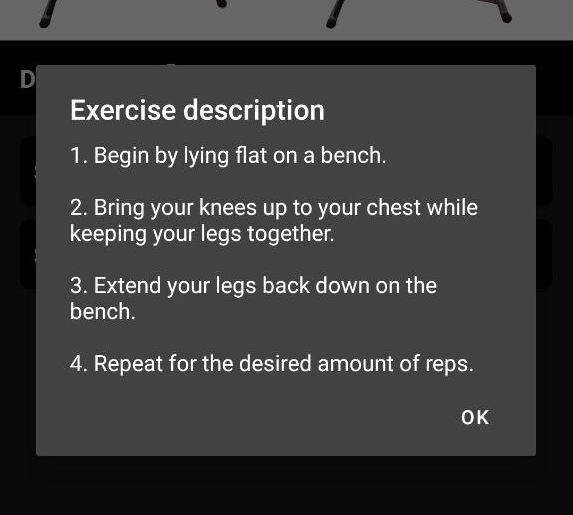
Treniņa izpildes logā (skat. 6.30. att.) tiek izvadīti treniņa un vingrinājuma nosaukumi un vingrinājuma kārtas numuru. Labajā-augšējā stūrī atrodas treniņa pārtraukuma poga, uzspiežot uz to, uz ekrāna izvadās treniņa pārtraukuma apstiprinājuma logs.

Zemāk atrodas bilde ar vingrinājuma pareizas tehnikas piemēru. Zem tā atrodas poga *Description (apraksts)*, uzspiežot uz to atvērās logs ar treniņa pareizas izpildes logu (skat. 6.31. att.).

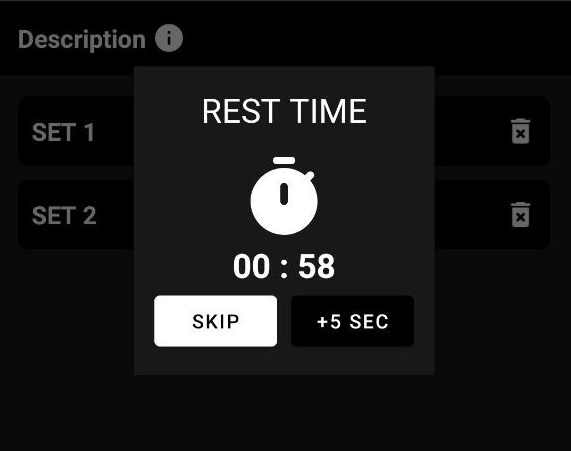
Pēc *Description* poga atrodas piegājiena saraksts, ar tā numuru, atkārtojuma un inventāra svara ievades laukiem. Gadījumā, kad vingrinājumu ir jāizpilda bez inventāra, tad svara ievades lauks netiek izvadīts. Katra piegājiena lauka beigās atrodas tas izdzēšanas poga, uzspiežot uz kuru, tas tiek dzēsts no saraksta.

Loga beigās atrodas pogas *Add Set (pievienot piegājienu)*, *Skip (izlaist)* un *Log Set (piegājiena pieraksts)*. Kad lietotājs uzspiež uz *Add Set* pogu, tiek pievienots jauns piegājiens. *Skip* poga izlaiž tekošu vingrinājumu, nesaglabājot to, ka pabeigto un izvada nākamo vingrinājumu. Kad lietotājs uzspiež uz *Log Set* pogu, piegājiena dati tiek saglabāti sistēmā. Kad visi piegājieni nav pabeigti, uz ekrāna tiek izvadīts starp piegājienu atpūtas taimeris (skat. 6.32. att.), ar divām pogām – *Skip (izlaist)* un laika pievienošana taimerim.

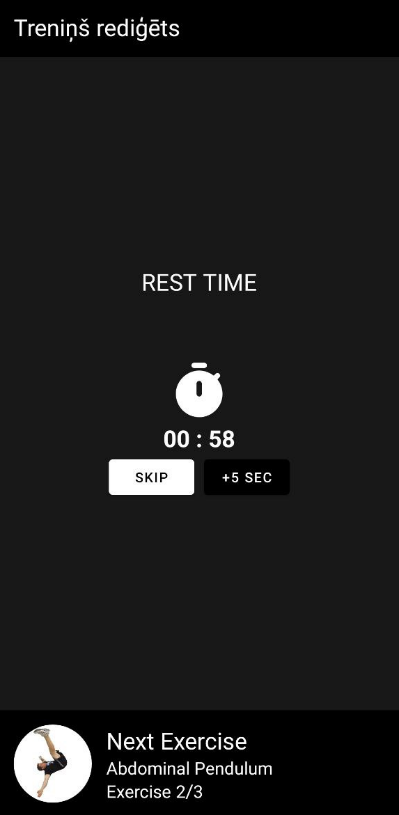
****Gadījumā, kad visi piegājieni ir pabeigti, uz ekrāna tiek izvadīts starp vingrinājumu taimera logs (skat. 6.33. att.). Logā tiek izvadīts treniņa nosaukums, atpūtas taimeris (līdzīgs tam, kas bija aprakstīts iepriekš) un, apakšā, nākamais vingrinājums.

6.30. att. Treniņa izpildes logs

6.31. att. Treniņa pareizas izpildes apraksta logs

****

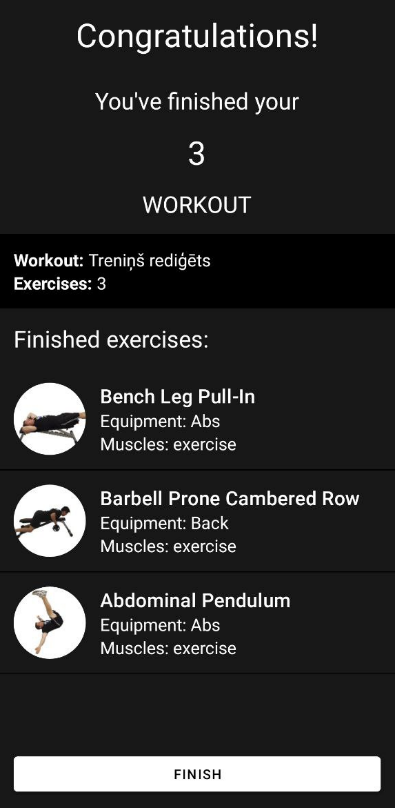
6.32. att. Starp piegājienu atputas taimeris

****

6.33. att. Starp vingrinājumu atputas taimeris

**Treniņa nobeiguma logs**

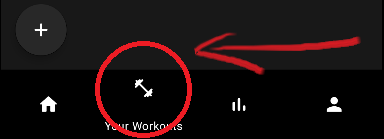
Kad lietotājs pabeidz treniņu, uz ekrāna tiek izvadīts treniņa nobeiguma logs (skat. 6.34. att.) ar pabeigto treniņu skaitu, izpildīta treniņa nosaukumu un tā vingrinājuma skaitu, un pabeigto vingrinājuma sarakstu.



6.34. att. Starp vingrinājumu atputas taimeris

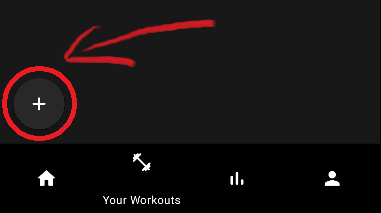
**6.4. Testa piemērs**

Testa piemēra nolūks ir parādīt lietotāja treniņa izveide – nosaukuma, vingrinājuma izveide un dzēšana.

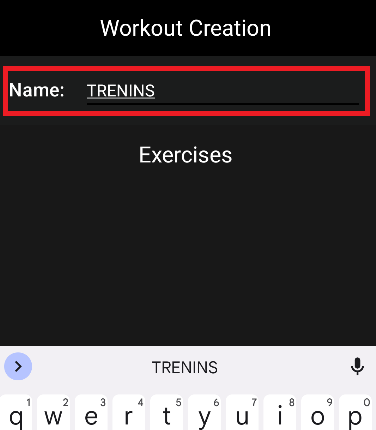
Pirmais solis uz vēlamā rezultāta sasniegšanu ir jāatver *Your Workouts* logu, uzspiežot uz attiecīgu pogu apakšizvēlnē (skat. 6.35. att.).

6.35. att. *Your Workouts* loga izvēle

Pēc loga izvēlēs uz ekrānā būs izvadīts lietotāja izveidotu treniņu sarakstu, bet, ja lietotājam nav izveidotu treniņu, būs izvadīta tekst “No workout has been created yet”, jeb “Nav izveidotu treniņu”. Lai izveidot jaunu treniņu ir jāuzspiež uz pogu kreisajā-zemajā stūrī (skat. 6.36. att.).



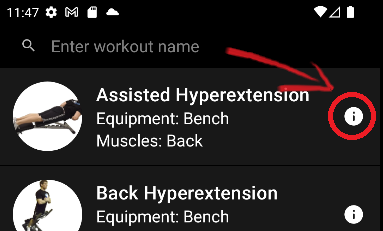
6.36. att. Lietotāja treniņa izveide

Pēc uz poga uzspiešanas, uz ekrānā izvadīs treniņa izveides logu. Augšā lietotājs var ievadīt treniņa nosaukumu (skat. 6.37. att.).

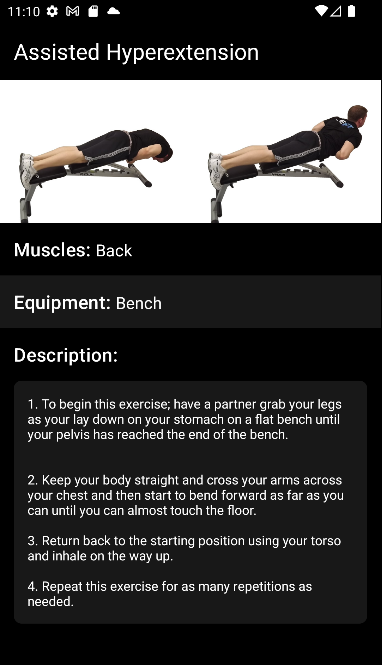
6.37. att. Lietotāja treniņa nosaukuma ievadē

Lai pievienot vingrinājumu ir jāuzspiež uz pogu kreisajā-zemajā stūrī (skat. 6.38. att).

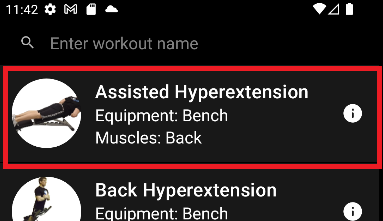
6.38. att. Vingrinājuma pievienošanā treniņam

Lai izvadīt vingrinājuma aprakstu, jāuzspiež uz pogu labajā pusē (skat. 6.39. un 6.40. att.).

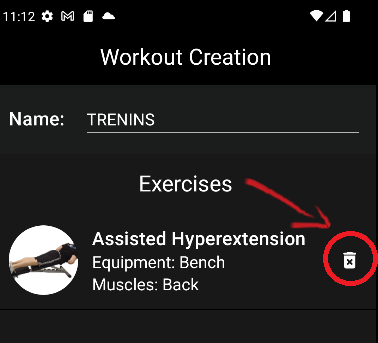
6.39. att. Vingrinājuma apraksta poga



6.40. att. Vingrinājuma apraksts

Lai pievienot vingrinājumu treniņam, jāuzspiež uz vingrinājumu no saraksta, kuru jūs gribat pievienot (skat. 6.41. att.).

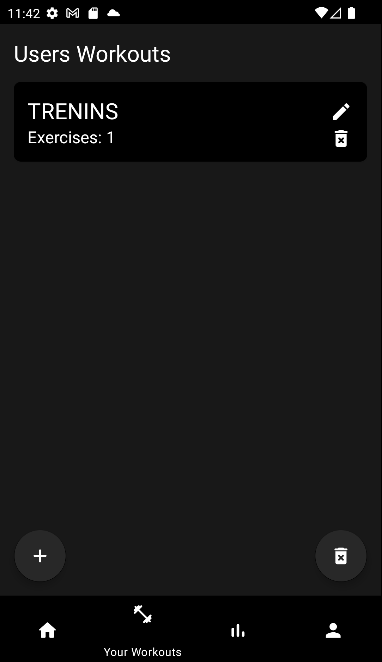
6.41. att. Vingrinājuma pievienošana treniņam

Lai izņemt vingrinājumu no treniņa ir jāuzspiež uz pogu, kas atrodas vingrinājuma rinda labajā pusē (skat. 6.42. att.).

6.42. att. Vingrinājuma dzēšana no treniņa

Lai saglabāt treniņu ir jāuzspiež uz pogu labā-zemajā stūrī, pēc pogas uzspiešanas uz ekrānā būs izvadīts lietotāja treniņu saraksts (skat. 6.43. un 6.44. att.).

6.43. att. Treniņa izveides apstiprināšana



6.44. Lietotāja treniņu saraksts

**NOBEIGUMS**

Darba sākumā galvenais mērķis bija realizēt mobilo lietotni, ar kuru būtu viegli un ērti sadarboties un, kas derētu gan tiem, kas tikai grib sākt savu ceļu sportā un fitnesā, gan cilvēkiem ar pieredzi šajā jomā. Es uzskatu, ka kvalifikācijas darbā izvirzītais mērķis ir sasniegts. Ir izveidota mobilo ierīču lietotne, kas dod iespēju lietotājam izveidot savus treniņus un izpildīt gan tos, gan jau iepriekš izveidotas treniņu programmas, ka arī saglabāt un apskatīt savus rezultātus.

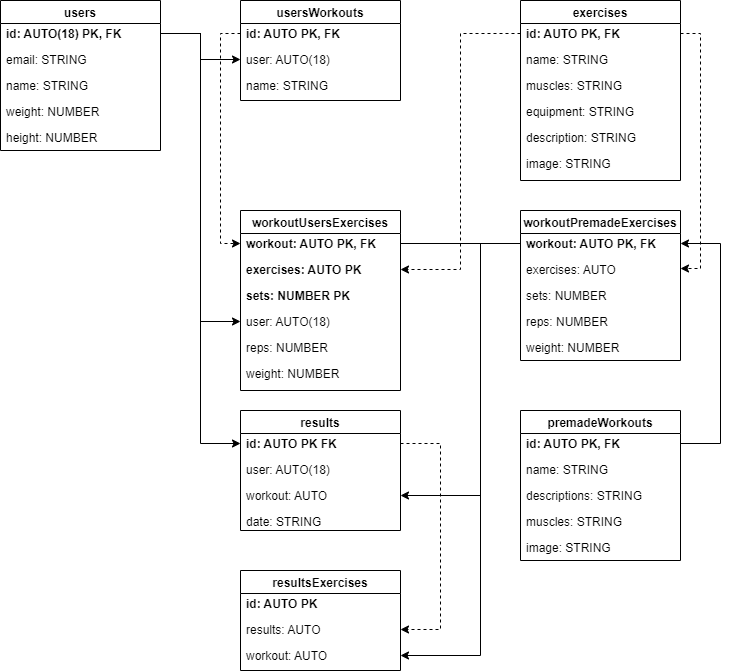
Neskatoties uz to, ka kvalifikācijas darba mērķis ir sasniegts, manuprāt, lietotni vēl ir jāattīsta. Un ir jāattīstās ne tikai lietotnei, bet arī man, ka programmētajam. Šis darbs ir mana pirmā pieredze mobilo ierīču, konkrētāk Android, lietotnes un NoSQL datu bāzes izstrādē. Projekta sākumā esmu saticis ar grūtībām, kas bija risināti un darba gaitā uzzināju daudz jaunas lietas. Es simtprocentīgi turpināšu projekta attīstīšanu, jo man ir diezgan lieli plāni uz to, ka arī tas palīdzes atbalstīt un pilnveidot savas prasmes programmēšanā.

**INFORMĀCIJAS AVOTI**

* Kotlin dokumentācija - <https://developer.android.com/kotlin/first> - (Resurss apskatīts 22.05.2022.);
* Firebase dokumentācija - <https://firebase.google.com/docs/guides> - (Resurss apskatīts 22.05.2022.);
* Android studio dokumentācija - <https://developer.android.com/docs> - (Resurss apskatīts 20.05.2022.);
* Algolia dokumentācija - <https://www.algolia.com/doc/> - (Resurss apskatīts 12.06.2022.);
* Glide bibliotēkas repozitorija - <https://github.com/bumptech/glide> - (Resurss apskatīts 31.05.2022.);
* Kotlin kurss - <https://www.youtube.com/watch?v=wuiT4T_LJQo&ab_channel=DonnFelker-FreelancingforSoftwareDevelopers> - (Resurss apskatīts 15.05.2022);
* Firebase kurss - <https://www.youtube.com/watch?v=hoK9OOP1KZw&ab_channel=tutorialsEU> – (Resurss apskatīts 18.05.2022.).

**PIELIKUMI**

1. pielikums

**Datu bāzes klāšu shēma**