

Profesionālās izglītības kompetences centrs

„Rīgas Valsts tehnikums”

Izglītības programma: Programmēšana

KVALIFIKĀCIJAS DARBS

Fizisko aktivitāšu mobilā aplikācija

Paskaidrojošais raksts 74 lpp.

Audzēknis:

Roberts Kristians Markūns

Vadītājs:

Igors Litvjakovs

Normu kontrole:

Olga Sabanska

RĪGA

2021

ANOTĀCIJA

Kvalifikācijas darba laikā tika izstrādāta fizisko aktivitāšu mobilā aplikācija. Darba autors ir PIKC Rīgas Valsts tehnikuma audzēknis Roberts Kristians Markūns. Darba vadītājs ir PIKC Rīgas Valsts tehnikuma skolotājs Igors Litvjakovs.

Kvalifikācijas darba izstrādāta mobila aplikācija, kas ir tēmēta uz lietotājiem, kuri vairākas stundas pēc kārtas pavada sēžot, tādējādi veicinot fiziskā stāvokļa pasliktināšanos, kas ietekmē cilvēka veselību, pašsajūtu un garastāvokli. Tas viss tiek pastiprināts, tā dēļ ka pasaule ir pandēmija, kas piespiež lielai daļai cilvēku sēdēt mājas un pavadīt vairāk stundu mazkustība nekā iepriekš. Ar aplikācijas palīdzību lietotājs var: likt atgādinājumus, lai izkustētos, atrast vingrinājumus, lai izstaipītos, pierakstīt uzdevumus, kas jāpaveic, meklēt vingrinājumus, priekš muskuļu attīstības un sameklēt vajadzīgo ekipējumu, lai atvieglotu vai izpildītu, kādu vingrinājumu. Priekš aplikācijas izstrādes tika izmantotas Java, PHP un XML valodas, kā arī MySQL priekš relāciju datubāžu vadības sistēmas.

Kvalifikācijas darba paskaidrojošais raksts sevī ietver ievadu, uzdevuma nostādni, prasību specifiku, uzdevuma risināšanas līdzekļu izvēles pamatojumu, programmatūras produkta modelēšanas un projektēšanas aprakstu, datu struktūru aprakstu, lietotāja ceļvedi, nobeigumu un informācijas avotus. Ievada ir aprakstīts iemesls kādēļ ir vajadzīga šāda aplikācija, par aplikācijas nosaukumu un ko tā spēs darīt. Uzdevuma nostādne ir kvalifikācijas darba izveidošanas mērķis un tā uzdevumi. Prasību specifika ir aprakstīta ieejas un izejas informācija, kā arī funkcionālas un nefunkcionālas prasības. Uzdevuma risināšanas līdzekļu izvēles pamatojuma apraksta, ko izmantoja, lai veidotu aplikāciju, kāpēc un versiju. Programmatūras produkta modelēšanas un projektēšanas apraksta ir redzamas datu plūsmas diagrammas, ER diagrammas un sistēmas arhitektūra un tas viss detalizēti aprakstīts. Datu struktūru apraksta ir tabulu struktūra un aprakstu, kā arī relāciju shēmu. Lietotāja ceļvedi ir aprakstīts kāds aparātūra un programmatūrā vajadzīga, ka palaist aplikāciju, visas funkcijas un ekrāni ar attēliem un testa piemērs. Nobeiguma, kas izveidots un ko ieguvu no šī darba.

Kopumā kvalifikācijas darba apjoms ir 74 lpp, kurā ietilpst 53 attēli un 10 tabulas.

ANNOTATION

During the qualification work, a mobile application for physical activities was developed. The author of the work is Roberts Kristians Markūns, a student of PIKC Riga State Technical School. The supervisor is Igors Litvjakovs, a teacher at PIKC Riga State Technical School.

A mobile application has been developed for the qualification paper, which is aimed at users who spend several consecutive hours sitting, thus contributing to the deterioration of their physical condition, which affects a person's health, well-being and mood. All this is exacerbated by the fact that the world has a pandemic that is forcing a large proportion of people to sit at home and spend more hours still than before. With the help of the application, the user can: set reminders to move, find exercises to stretch, write down tasks to be performed, search for exercises, for muscle development and find the necessary equipment to facilitate or perform an exercise. Java, PHP and XML languages were used for the application development, as well as MySQL for relational database management systems.

The explanatory article of the qualification paper includes an introduction, task statement, requirements specification, justification of choice of task solving means, description of software product modeling and design, description of data structures, user guide, conclusion and information sources. The introduction describes the reason why such an application is needed, the name of the application and what it will be able to do. The task statement is the aim of creating a qualification paper and its tasks. The requirements specification describes the input and output information as well as the functional and non-functional requirements. A description of the rationale for the choice of problem-solving tools used to build the application, why, and version. The software product modeling and design description shows the data flow diagrams, ER diagrams and system architecture and all this is described in detail. Data structure description is a table structure and description, as well as a relational scheme. The user guide describes what hardware and software is needed to run the application, all the features and screens with images and a test example. The final created and gained from this work.

In total, the amount of qualification work is 74 Pages long that includes 53 pictures and 10 tables.

SATURS

IEVADS	5
1. UZDEVUMA NOSTĀDNE.....	6
2. PRASĪBU SPECIFIKĀCIJA.....	7
2.1. Ieejas un izejas informācijas apraksts	7
2.1.1. Ieejas informācijas apraksts.....	7
2.1.2. Izejas informācijas apraksts.....	8
2.2. Funkcionālās prasības.....	9
2.3. Nefunkcionālās prasības.....	12
3. UZDEVUMA RISINĀŠANAS LĪDZEKĻU IZVĒLES PAMATOJUMS.....	14
4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA	19
4.1. Sistēmas struktūras modelis	19
4.1.1. Sistēmas arhitektūra	19
4.1.2. Sistēmas ER modelis.....	21
4.2. Funkcionālais sistēmas modelis	23
4.2.1. Datu plūsmu modelis	23
5. DATU STRUKTŪRAS APRAKSTS.....	30
6. LIETOTĀJU CEĻVEDIS	35
6.1. Sistēmas prasības aparatūrai un programmatūrai	35
6.2. Sistēmas instalācija un palaišana.....	38
6.3. Programmas apraksts.....	45
6.4. Testa piemērs.....	65
NOBEIGUMS	72
INFORMĀCIJAS AVOTI.....	73

IEVADS

Kvalifikācijas darba mērķis ir izveidot ērtu un kvalitatīvu mobilo aplikāciju, ar kuras palīdzību strādājot mājas apstākļos cilvēks var palikt aktīvs kā arī attīstīt savu fizisko formu, zaudēt svaru vai arī vienkārši izstaipīties. Lietotājs varēs reģistrēties, apskatīt vingrinājumus, apskatīt ekipējumu un to iegādāties, uzstādīt laiku vingrinājumam vai uzņemt atpūtas laiku, varēs uzstādīt vingrinājuma izpildes reižu ģenerētājs, kurš vadoties no lietotāja mazāka ievadīta vingrinājuma reižu skaita un lielāka vingrinājuma reižu skaita izvadīs uzģenerētu vingrinājumu izpildes reižu skaitu, lietotājs varēs uzlikt atgādinājumu, lai ik pēc stundas atietu no darba vietas un izkustētos, kā arī lietotājs varēs izveidot sev pierakstus par treniņiem vai lietām, ko vēlās paveikt tāds kā 'ToDo' saraksts jeb darāmo lietu saraksts, kuriem var uzstādīt datumu uz kuru ir jāizpilda un arī atgādinājumu.

Kvalifikācijas darba ietvaros izstrādāta aplikācija 'sthenos', kuras nosaukums ir sengrieķu vārds, kas nozīmē "spēks"(liels garīgais spēks, drosme, apņēmība), nodrošina visu nepieciešamo, lai lietotājs varētu viegli un uzskatāmi atrast vajadzīgos vingrinājumus, skaidrojumus kā pareizi tie būtu jāveic, kopā ar grūtības pakāpi un kādas muskuļu grupas tiks izmantotas. Tas viss sakomplektēts moderna izskata un ērti lietojamā veidā, lai lietotājam pēc iespējas mazāk sagādātu kādas grūtības vai nesaprašanas aplikācijas lietošanas laikā. Lietotāja pieredze ir viena no svarīgākajām lietam tādēļ aplikācijas interfeiss ir ļoti moderni veidots, kas uzlabotu pieredzi ar aplikāciju.

1.UZDEVUMA NOSTĀDNE

Kvalifikācijas darba uzdevums ir izveidot vingrinājumu mobilo aplikāciju, kuru var izmantot tikai reģistrēts lietotājs. Ar sistēmas palīdzību varēs viegli atrast vingrinājumus ekipējumu un citus noderīgus rīkus.

Mobilās aplikācijas lietotnei ‘sthenos’ ir jāizpilda vairākas funkcionalitātes:

- ❖ lietotāja profilā veidošana;
- ❖ hronometra izmantošana;
- ❖ vingrinājuma izpildes skaita ģenerēšana;
- ❖ vingrinājuma apskate;
- ❖ ekipējuma apskate;
- ❖ ‘ToDo’ jeb Darāmo lietu saraksts;
- ❖ atgādinājuma uzstādīšana;
- ❖ lietotājs varēs ātri pārvietoties cauri aplikācijai ar navigācijas joslu;
- ❖ termiņa uzstādīšana.

2. PRASĪBU SPECIFIKĀCIJA

2.1. Ieejas un izejas informācijas apraksts

2.1.1. Ieejas informācijas apraksts

Sistēmā tiks nodrošināta šāda ieejas informācijas apstrāde.

- ❖ Informācija par lietotāju, kas sastāv no:
 - cilvēka pilns vārds, kas ir teksts, neierobežots;
 - lietotājvārds, kas ir simbolu virkne, ne garāka par 50;
 - paroles, kas ir teksts, neierobežots;
 - e-pasta, kas ir simbolu virkne, ne garāka par 50.
- ❖ Informācija par vingrinājumiem, kas sastāv no:
 - nosaukuma, kas ir simbolu virkne, ne garāka par 256;
 - apraksts, kas ir teksts, neierobežots;
 - bildes, kas ir attēls, neierobežots;
 - grūtības pakāpe, kas ir simbolu virkne, ne garāka par 32;
 - muskuļu grupa, kas ir teksts, neierobežots.
- ❖ Informācijā par grupām, kas sastāv no:
 - nosaukuma, kas ir simbolu virkne, ne garāka par 50.
- ❖ Informācija par ekipējumu, kas sastāv no:
 - nosaukuma, kas ir simbolu virkne, ne garāka par 256;
 - bildes, kas ir attēls, neierobežots;
 - reitinga, kas ir vesels skaitlis un kura vērtība ir no 1 līdz 5;
 - cena, kas ir decimāls skaitlis, neierobežots;
 - apraksts, kas ir teksts, neierobežots.
- ❖ Informācija par 'ToDo', kas sastāv no:
 - apraksts, kas ir teksts, neierobežots;
 - izpildes termiņš, kas ir sqlite datums.
- ❖ Informācija par vingrinājumu izpildes skaitu ģenerators, kas sastāv no:
 - minimālais skaitlis, kas ir vesels skaitlis un kura vērtība ir no 0 līdz 100;

- maksimālais skaitlis, kas ir vesels skaitlis un kura vērtība ir no 0 līdz 100.
- ❖ Informācija par paziņojumu, kas sastāv no:
 - laiks (kurā nostrādās paziņojums), kas ir laiks 24 stundu apmērā.

2.1.2. Izejas informācijas apraksts

Sistēma nodrošinās šādas informācijas izeju :

- ❖ Vingrinājumu izvade. Katru reizi, kad tiks atvērta kāda no vingrinājuma aktivitātēm tiks parādīta: vingrinājuma bilde, vingrinājuma nosaukums, vingrinājuma grūtības pakāpe, vingrinājuma muskuļu grupa un vingrinājuma apraksts. Šāda veida apkopojot visu informāciju, kas lietotājam jāzina, lai veiktu konkrēto vingrinājumu.
- ❖ Vingrinājumu izpildes skaitu ģenerators rezultāts. Šeit lietotājam ievadot - mazāko vingrinājuma izpildes reižu skaitu un lielāko vingrinājuma izpildes reižu skaitu - programma izvadīs jebkādu skaitli dotajā intervālā, kuru lietotājs var izmantot, lai veiktu vingrinājumu pēc izvades reižu skaita.
- ❖ ‘ToDo’ izvade. Šeit lietotājam būs pieejami visi viņa veidotie pieraksti jeb lietas, kas jāizdara konkrēta dienā, kuru ir norādījis pats lietotājs. Pierakstos izvada aprakstu un datumu, kurā ir jāpaveic konkrētais pieraksts. Šie pieraksti tiek grupēti 3 grupās ‘today’ jeb šodien, ‘tomorrow ’ jeb rītdien un ‘upcoming’ jeb gaidāmās lietas, kas ir jāizdara lietotājam. Lietotājs savus pierakstus var rediģēt un dzēst.
- ❖ Paziņojuma izvade. Šeit lietotājs varēs uzstādīt sev atgādinājumu par to ka ir jāpieceļas un jāizkustas no darba vietas izvēlētajā/ievadītajā laikā.
- ❖ Ekipējuma izvade. Katru reizi, kad tiks atvērta kāda no ekipējuma aktivitātēm tiks parādīta: ekipējuma bilde, ekipējuma nosaukums, ekipējuma cena un ekipējuma apraksts. Šāda veida apkopojot visu informāciju, kas lietotājam jāzina, par ekipējumu un tā iegādi.
- ❖ Vingrinājumu grupu izvade. Šeit tiks izvadītas visas grupas un to nosaukumi. Izvēloties kādu grupu atvērsies jauna aktivitāte, kur būs vingrinājumi no konkrētas grupas, kurus varēs apskatīt pēc vingrinājuma izvades.

2.2. Funkcionālas prasības

Sistēmas darbībai tiek izvirzītas šādas funkcionālās prasības.

- ❖ Lietotāja reģistrācija:
 - Obligātie un neobligātie lauki reģistrācijas procesā. Pārbaudes katra lauka vai ir aizpildīts;
 - Paroles šifrēšana;
 - Lietotājvārds un e-pasts nedrīkst sakrist ar datubāzē jau esošiem datiem, tiem ir jābūt unikāliem ierakstiem.
- ❖ Lietotāja profila iespējas:
 - Iziešana no profila.
- ❖ Grupas:
 - Sastāv no 4 grupām: mobilitātes vingrinājumu grupa, iekšdurvju vingrinājumu grupa, ārdurvju vingrinājumu grupa un ekipējuma grupa;
 - Katrā grupā ir vairākas vingrinājumu apakšgrupas, kas sadala vingrinājums pēc vingrinājuma veida;
 - Katram vingrinājumam ir grūtības pakāpes grupa;
 - Katram vingrinājumam ir muskuļu grupa;
 - Pierakstu grupēšana 3 grupas: šodien, rītdien, gaidāmie.
- ❖ Vingrinājumi:
 - Vingrinājuma apskate;
 - Poga, kuru nospiežot pāriet uz hronometru un atrodas tikai mobilitātes vingrinājumu grupa, iekšdurvju vingrinājumu grupa un ārdurvju vingrinājumu grupa ;
 - Poga, kuru nospiežot pāriet uz vingrinājuma izpildes reižu ģenerators un atrodas tikai iekšdurvju vingrinājumu grupa un ārdurvju vingrinājumu grupa;
 - Lidojoša navigācijas josla;
 - Ritinot, vingrinājuma aktivitāti uz augšu, vingrinājuma bilde samazinās un pazūd, mainoties uz krāsu, kas atbilst grupā kurā atrodas vingrinājums. Ja vingrinājums atrodas mobilitātes grupā tad krāsa ir sarkana, ja iekšdurvju

vingrinājumu grupa tad krāsa ir dzeltena, bet ja ārdurvju vingrinājumu grupa tad krāsa ir gaiši zaļā krāsa.

❖ Ekipējums:

- Ekipējuma apskate;
- Poga, kuru nospiežot ekipējumu pievieno grozam, kas izvada paziņojumu ka ekipējums ir pievienots grozam;
- Lidojoša navigācijas josla – navigācijas josla nav pievienota neviena ekrāna malai un seko līdz lietotāja skatu punktam aktivitātē;
- Ritinot ekipējuma aktivitāti uz augšu, ekipējuma bilde samazinās un pazūd, mainoties uz sarkanu krāsu.

❖ Navigācija:

- Sānu navigācija pirmā daļa, kas atrodas sākuma lapā, kur ir iespējams izvēlēties: sākuma lapu, mobilitātes vingrinājumu grupu, iekšdurvju vingrinājumu grupu, ārdurvju vingrinājumu grupu, ekipējuma grupu;
- Sānu navigācija otra daļa, atgādinājuma aktivitāti un pierakstu aktivitāti;
- Sānu navigācija trešā daļa Dalīšanas izvēle, kas izvada ziņojumu, kad ir izvēlēts un novērtē mūs izvēle, kas izvada ziņojumu, kad ir izvēlēts;
- Lidojoša navigācijas josla, kas atrodas katrā vingrinājuma aktivitātē, katrā ekipējumā aktivitātē, atgādinājumu aktivitātē un pierakstu aktivitātē;
- Lidojoša navigācijas josla sastāv no 5 izvēles iespējam: sākuma lapu, mobilitātes vingrinājumu grupu, iekšdurvju vingrinājumu grupu, ārdurvju vingrinājumu grupu, ekipējuma grupu;
- Lidojoša navigācijas josla nav pievienota nevienai ekrāna malai un seko līdz lietotāja atrašanās vietai uz ekrāna pat, ja lietotājs ritina ekrānu uz augšu vai leju.

❖ Hronometrs:

- Pogā, kas uzsāk laika skaitīšanu;
- Pogā, kas beidz laika skaitīšanu;
- Laika attēlošana uz ekrāna;
- Pulksteņa animācijas uzsākšana un beigšana.

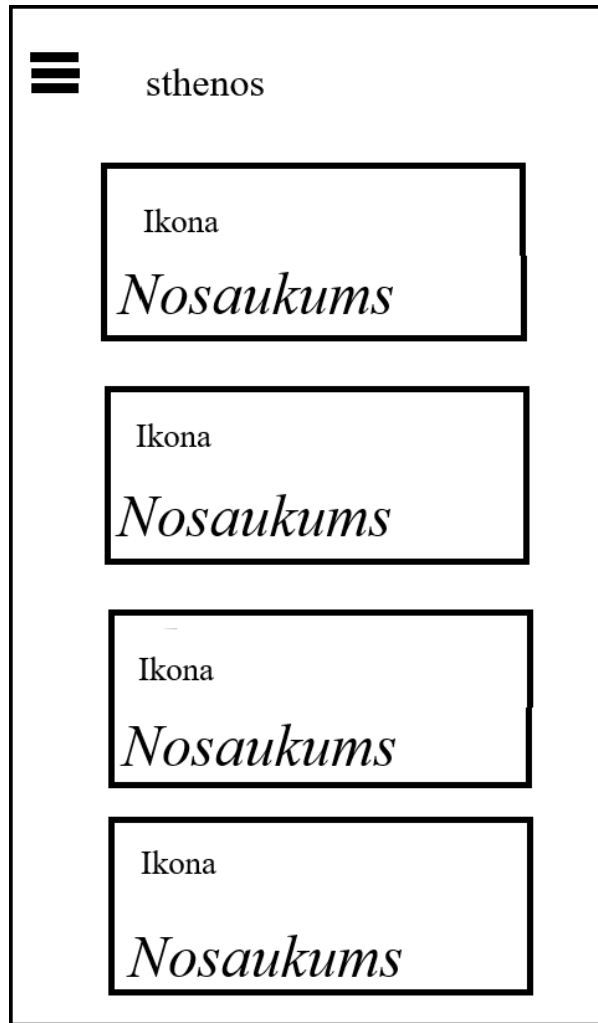
❖ Pieraksti:

- Pierakstu pievienošana, kur ievada aprakstu un izvēlas izpildes datumu, kā arī pēc izvēles pievienot atgādinājumu;
 - Pierakstu labošana, datumu un aprakstu;
 - Pierakstu dzešanā;
 - Pierakstu grupēšana pēc izpildes datuma. Ja izpildes datums ir šodien tad tiks pievienots šodienas grupai, ja izpildes termiņš ir rītdiena tad tiks pievienots rītdienas grupai, bet, ja izpildes termiņš ir aiznākama diena vai vēl tālāk tad tiks pievienots gaidāmai grupai;
 - Lidojoša navigācijas josla;
 - Pierakstu pārbaude, ja nav pievienots neviens pieraksts vai visi uzdevumi jau ir paveikti tad uz pierakstu aktivitātes rādās paziņojums, ka nav neviens pieraksts.
- ❖ Vingrinājuma izpildes reižu ģenerators:
- Poga, kas pēc mazākas iespējams vērtības un lielākas iespējamās vērtības nejauši izvēlas skaitli mazāka un lielāka skaitļa intervālā priekš vingrinājuma izpildes reižu skaita;
 - Rezultāta atspoguļošana uz ekrāna.
- ❖ Atgādinājumi:
- Pogā, kas atver laika izvēli;
 - Lika izvēle: stundās un minūtes;
 - Atgādinājuma laika atspoguļošana uz ekrāna;
 - Atgādinājuma pārbaude, kas pārbauda vai ir uzstādīts atgādinājums, ja nav tad atgādinājuma aktivitāte rādās paziņojums, ka nav uzstādīts atgādinājums;
 - Pogā, kas atceļ atgādinājumu.

2.3. Nefunkcionālas prasības

Sistēmas darbībai tiek izvirzītas šādas nefunkcionālas prasības.

- ❖ Sistēmas saskares valodai ir jābūt angļu valodai.
- ❖ Mobilai aplikācijai jābūt savietojamai ar jebkuras mobilas ierīces ekrānu.
- ❖ Aplikācijai jābūt ātram ielādēs un apstrādes laikam, un tā nedrīkst pārslogot mobilas ierīces procesoru.
- ❖ Aplikācijai ir jābūt maksimāli optimizētai, lai neaizņemtu daudz atmiņas vietas.
- ❖ Aplikācijai ir jābūt moderna izskata un lietotāja interfeisam jābūt gludām: animācijām, pārvietošanos caur aplikāciju un tās funkcijām.
- ❖ Aplikācijas fona krāsai ir jābūt baltai, bet akcenta krāsai ir jābūt sarkanā.
- ❖ Visiem lapas galvenajiem virsrakstiem jābūt ieturēt vienādā stilā.
- ❖ Tekstam jābūt 'Rubik' fontā.
- ❖ Pogām jābūt ar noapaļotiem galiem un krāsa ir atkarīga no vingrinājuma grupas vai aktivitātes.
- ❖ Mobilitātes grupas primārai krāsai jābūt sarkanai, fona krāsai ir jābūt baltai.
- ❖ Iekšdurvju vingrinājumu grupas primārai krāsai jābūt oranžai un dzeltenai, fona krāsai ir jābūt baltai.
- ❖ Ārdurvju vingrinājumu grupas primārai krāsai jābūt zaļai, fona krāsai ir jābūt baltai.
- ❖ Ekipējuma primārai krāsai jābūt melnai un sarkanai, fona krāsai ir jābūt baltai.
- ❖ Lidojoša navigācijas joslai fona krāsai ir jābūt melnai un ikonām jābūt baltām.
- ❖ Pogām, kas nav saistītas ar kāda vingrinājuma grupu ir jābūt violetas un sarkanas krāsas gradientam.
- ❖ Katrai vingrinājuma grupai un ekipējumam ir sava piešķirta krāsa. Galvenās grupas ir sadalītas vertikāli, bet apakšgrupas ir sadalītas horizontāli divās rindās ar apakšgrupas ikonu, nosaukuma un vingrinājuma grupas krāsu kā fonu.
- ❖ Sākuma lapas navigācijas joslai ir jābūt izvietotam ekrāna augšējā kreisajā stūrī (skat. 2.3.1.att.).



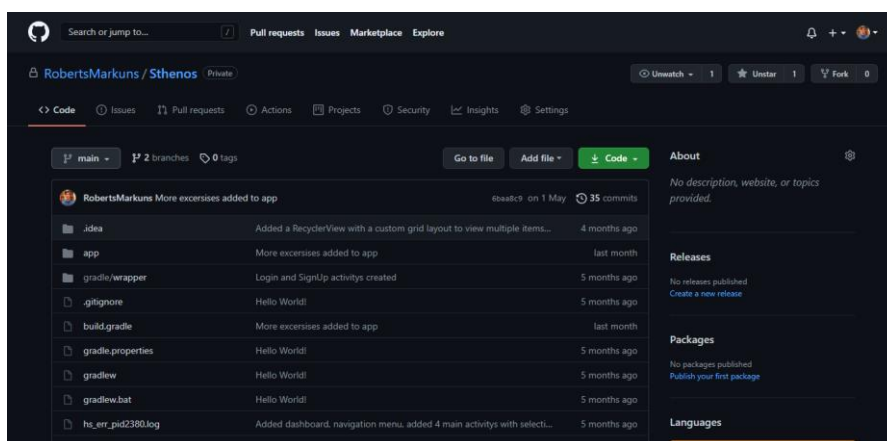
2.3.1.att. Sakuma lapas skice

3. UZDEVUMA RISINĀŠANAS IZVĒLES PAMATOJUMS

Kvalifikācijas darba ietvaros tika izmantotas vairākas atklātā pirmkoda skriptu valodas. Pamat valoda, kas izmantota aplikācijas veidošanai ir klases balstīta un objektorientēta programmēšanas valoda Java. Šo valodu izvēlējās, jo tā ir pamat valoda uz kuras būvē Android aplikācijas. Kvalifikācijas darbā šī valoda atbild par aplikācijas funkcionālo darbību. Izmantota tika Java SE8 (1.8.0_231) versija.

Lai izveidotu aplikācijas lietotāja interfeisu tika izmantota iezīmēšanas valoda XML. Izmantota tika XML 1.0 versija. Kvalifikācijas darbā šo valodu izmantoja, lai veidotu vairākus failu veidus. Izkārtotuma failus, kas atbild par pašu aktivitāšu izkārtotumā stilu. Manifest failu, kurus izmanto, lai definētu visas mūsu lietojumprogrammas sastāvdaļas. String failu, kurš aizstāj cietā kodējuma teksta virknes ar vienu teksta virkni. Zīmējuma failus, kuru izmanto nodrošina dažādas grafikas lietojuma elementiem vai skatiem. Krāsu failu, kurā glabā visas definētas krāsas ko izmanto aplikācijā.

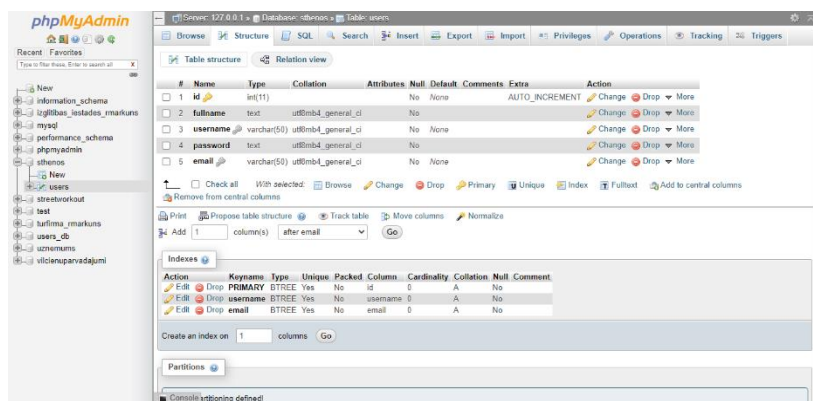
Aplikācijas versiju kontrolei tika izmantota Git sistēma. Šo sistēmu izvēlējās, jo tā ir mūsdienu standarts projektu versiju kontrolē, kā arī ir viegli uztverama un parocīga tīmekļa saskare GitHub (skat. 3.1. att.), kurā var apskatīt projektus un to versijas. Kvalifikācijas darba izmantoju Git un GitHub, lai saglabātu katru jauni veidoto funkciju vai interfeisa jaunu stila ieviešanu aplikācijā. Izmantota tika Git 2.24.1.windows.2 versija.



3.1. att. Tīmekļa saskarne GitHub sistēmai priekš projekta versijas kontroles.

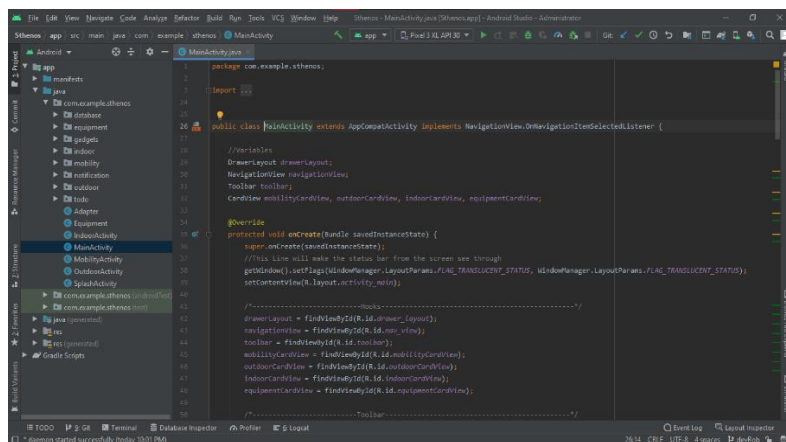
Lai tiktu galā ar servera pusi tika izmantota programmēšanas valoda – PHP. Šo valodu izvēlējās, jo tā ir mūsdienu standarts un tā ir no platformas neatkarīga valoda. Izmantota tika PHP 7.3.12 versija. Kvalifikācijas darbā tā tika izmantota, lai lietotāja reģistrācijas datus saglabātu un pārvietotu no aplikācijas uz datu bāzi.

Darbībām ar datubāzi tika izvēlēts atvērtā pirmkoda tīmekļa pielikuma rīks phpMyAdmin (skat. 3.2. att.), kurš ir paredzēts MySQL datubāžu pārvaldīšanai izmantojot tīmekļa saskarni. Šo iezīmēšanas valodu izvēlējās, jo tā ir ērti lietojama, draudzīga saskare ar lietotāju un ļauj pārvaldīt datubāzi bez tiešas SQL komandu ievades. Kvalifikācijas darbā tā tik izmantota, lai glabātu lietotāja datus. Izmantota tika phpMyAdmin 4.9.2 versija un MySQL 5.0.12 versija.



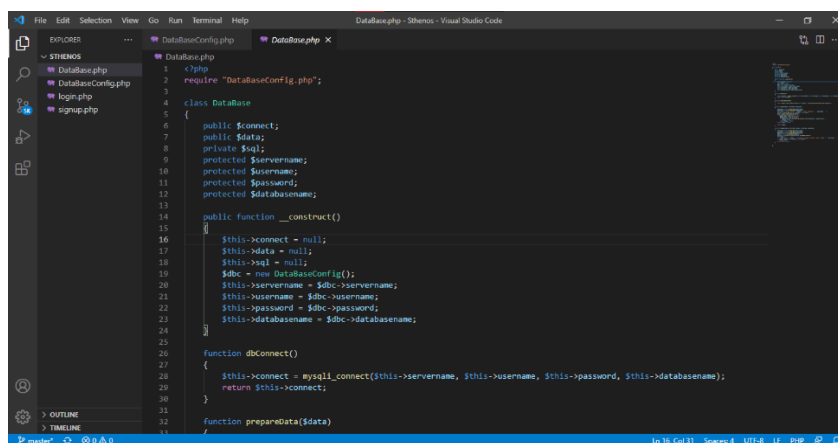
3.2.att. phpMyAdmin atvērtā pirmkoda tīmekļa pielikuma rīks

Aplikācijas kodēšanai, ar iepriekšminētajam programmēšanas valodām un sistēmām, ir nepieciešama izstrādes vide. Priekš Android mobilās aplikācijas kodēšanas es izmantoju Android Studio (skat. 3.3. att.), jo tā ir oficiālā integrētā Google Android operētājsistēmas izstrādes vide, kas veidota uz JetBrains IntelliJ IDEA programmatūras un paredzēta īpaši Android izstrādei. Izmantota tika Android Studio 4.1.3 versija.



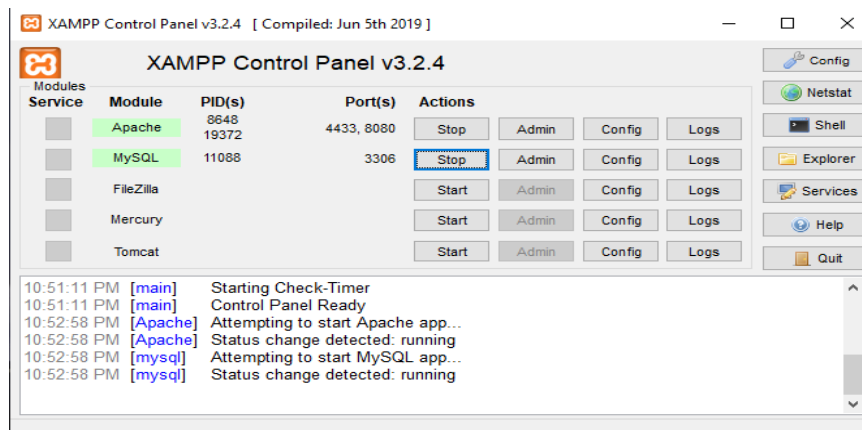
3.3. att. Android Studio mobilo aplikāciju izstrādes vide

Servera puses kodēšanai ar PHP es izmantoju Visual Studio Code. Visual Studio Code ir bezmaksas programmatūras pirmkodu redaktors, ko Microsoft izveidojis operētājsistēmām Windows, Linux un macOS. Šo izstrādes vidi izvēlējos, jo tā ir ļoti vienkāršā un visu nepieciešamo var lejupielādēt kā plugin jeb moduli, kā arī tas neaizņem daudz atmiņas vietas un Gits ir iebūvēts redaktora iekša. Izmantota tika Visual Studio Code 1.56.2 versija.



3.4. att. Visual studio code bezmaksas programmatūras pirmkodu redaktors

Priekš savienošanas ar lokālo serveri es izmantoju XAMPP (skat. 3.5. att.). XAMPP ir bezmaksas un atvērta koda starp platformu tīmekļa serveru risinājumu kaudzes pakete, kas galvenokārt sastāv no Apache HTTP servera, MariaDB datu bāzes un PHP rakstītu skriptu tulku. Kvalifikācijas darba izstrādes laikā izmantoju XAMPP, lai veidotu datubāzi phpMyAdmin uz lokālā servera un izmantoju XAMPP, lai savienotu lokālo serveri ar phpMyAdmin izmantojot Apache un MySQL. Izmantota tika XAMPP 3.2.4 versija.



3.5. att. XAMPP savienošanas ar lokālo serveri izmantojot Apache un MySQL.

Bilžu un Ikonu importēšanai izmantoju moduli Android Drawable Importer Plugin. Kvalifikācijas darbā šo moduli es izvēlējos izmantot, lai atvieglotu un paātrinātu bilžu un ikonu importēšanu Android Studio, kuras savādāk nevarētu vai sarežģītu importēšanu. Android Drawable Importer Plugin oriģinālais autors bija pārtraucis atjaunināt šo moduli tādēļ tas nestrādāja ar jaunāko Android studio versiju un nācās izmantot cita autorā dakšotu versijas atjauninājumu no GitHub (skatīt informācijas avotus 1 un 2). Izmantota tika pielietota šī moduļa pēdējā versija.

4. PROGRAMMATŪRAS PRODUKTA MODELĒŠANA UN PROJEKTĒŠANA

4.1. Sistēmas struktūras modelis

4.1.1. Sistēmas arhitektūra

Kopumā sistēma sastāvēs no 2 daļām

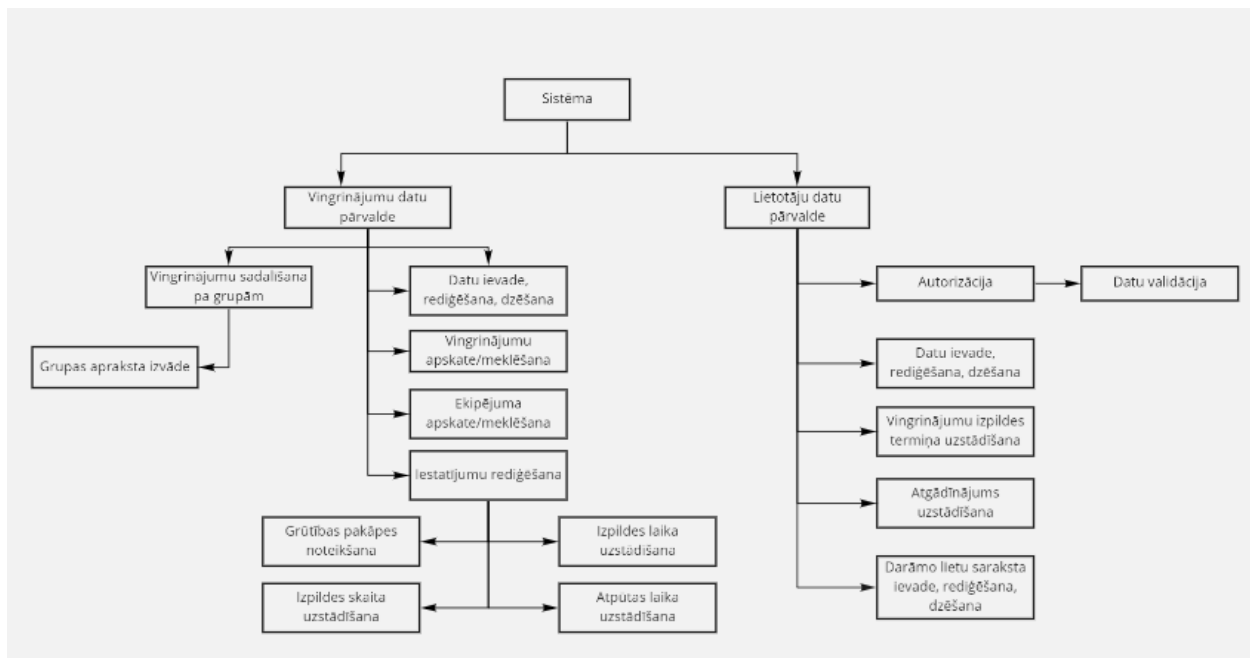
1. Lietotāja daļa.
2. Vingrinājumu daļa

Lietotāja daļa sastāv no visām lietotāju plānotajām funkcijām. Lietotāja funkcijas dalās autorizācija, lietotāja datu pārvaldības, vingrinājumu izpildes termiņa uzstādīšana un atgādinājuma uzstādīšana. Lietotāja autorizācija sastāv no lietotāju reģistrēšanās sistēmā un Datu validācijas, lai izmantotu sistēmu. . Lietotāja datu pārvalde sastāv no lietotāja funkcijām, kur lietotājs var pats izmainīt savus datus, piemēram, lietotāj vārdu. Lietotāja vingrinājumu izpildes termiņa uzstādīšana, spēj uzstādīt sev rakstisku atgādinājumu ar to, kas ir jāveic līdz kādam noteiktam laikam, tādu kā darāmo lietu sarakstu un papildus var uzlikt atgādinājumu, lai neaizmirstu. Lietotāja atgādinājuma uzstādīšana ir iespējams uzstādīt atgādinājumu, lai pieceltos no darba vietas un izkustētos ik pēc stundas.

Vingrinājumu daļa ir visas funkcijas, kuras saistītas no vingrinājumiem. Vingrinājumu funkcijas dalās vingrinājumu sadalīšana pa grupām, vingrinājumu datu ievade, rediģēšana un dzēšana, vingrinājumu meklēšana, ekipējuma apskate un meklēšana un iestatījumu rediģēšana. Vingrinājumu sadalīšana pa grupām vēl iedalās grupas apraksta izvade, kas ir ‘Mobility’ , ‘Indoor’ un ‘Outdoor’ grupas, kas sevī katra ietver vēl 8 grupās, kurās atrodas vingrinājumi atbilstoši grupas nosaukumam. Vingrinājumu datu ievade, rediģēšana un dzēšana ir vingrinājumu izveide, kur ielikto vingrinājuma bildi, vingrinājuma grūtības pakāpi, vingrinājuma muskuļu attīstošās grupas un vingrinājuma aprakstu var izveidot vingrinājuma lapu vai labot kādu no laikiem, piemēram, apraksts vai dzēst kādu no vingrinājuma lapām. Vingrinājumu meklēšana ir spēja viegli pārvietoties pa aplikāciju, lai viegli atrastu nepieciešamo vingrinājumu pēc vingrinājuma grupas nosaukuma, piemēram, ‘Pull-ups’. Ekipējuma apskate un meklēšana ir iespēja apskatīt nepieciešamo ekipējumu priekš katra vingrinājuma un, cik tas izmaksās

lietotājam, apskatot ir iespējams redzēt attēlu, aprakstu un cenu. Iestatījumu rediģēšana iedalās četrās daļās: izpildes laika uzstādīšana, atpūtas laika uzstādīšana, izpildes skaita uzstādīšana un grūtības pakāpes noteikšana. Izpildes laika uzstādīšana lietotājs spēj uzņemt laiku, cik ilgā laikā viņš veic, kādu vingrinājumu, kas palīdz lietotājam redzēt precīzu laiku, kas nepieciešams rezultāta sasniegumam. Atpūtas laika uzstādīšana ir lietotāja iespēja uzņemt noteiktu laiku priekš atpūtas starp vingrinājumiem. Izpildes skaita uzstādīšana dod lietotājam iespēju ievadīt mazāko iespējamo un lielāko iespējamo reižu skaitu, kas izvadīs uzģenerētu skaitli dotajā intervāla, kas būs lietotājam jāveic. Grūtības pakāpes noteikšana ir ievietota katrā vingrinājumā, lai lietotās zinātu, cik spējīgam viņam jābūt, lai veikto izvēlēto vingrinājumu.

Sistēmas arhitektūras diagrammu var apskatīt zemāk:



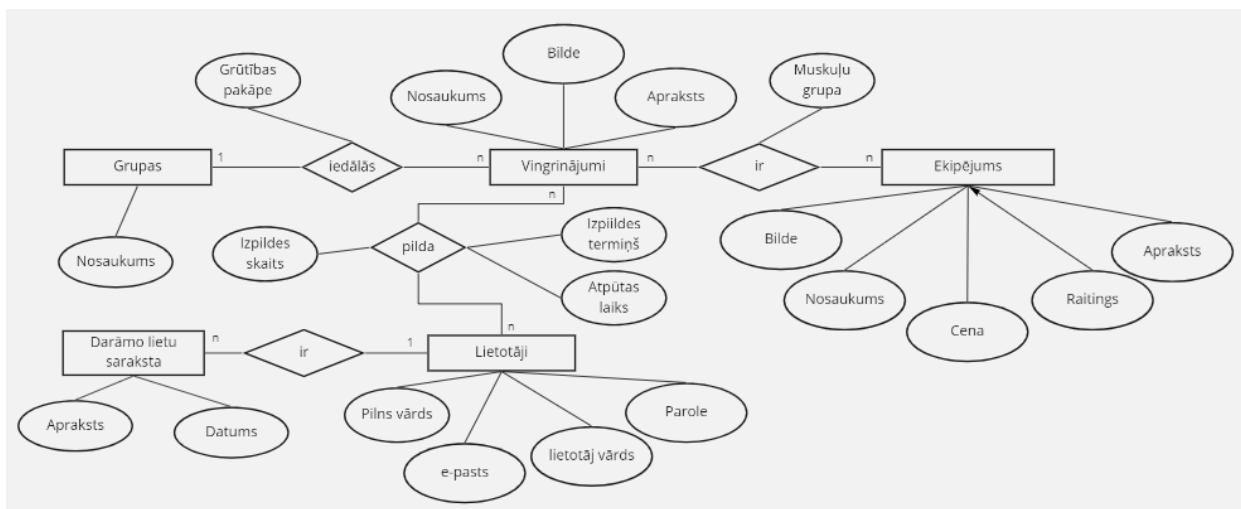
4.1. att. Sistēmas arhitektūra

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).

Datu bāzes ER modelis, kas ir dots zemāk, sastāv no sekojošām entītijām:

- **“Vingrinājumi”** – Apraksta visus vingrinājumus sistēma. Tās atribūtu kopums sevī ietver vingrinājumu aprakstu, vingrinājumu bildi un vingrinājumu nosaukumu.
- **“Grupas”** - Apraksta visas grupas sistēma. Tās atribūtu kopums sevī ietver grupas nosaukumu.
- **“Lietotāji”** - Apraksta visus lietotājus sistēma. Tās atribūtu kopums sevī ietver, lietotāja pilnu vārdu (vārds, uzvārds, un ja ir tad arī otrais vārds), lietotāja e- pasts, lietotāj vārds un lietotāja parole.
- **“Ekipējums”** - Apraksta visu ekipējumu sistēma. Tās atribūtu kopums sevī ietver ekipējuma bildi, ekipējuma nosaukumu, ekipējuma reitings, ekipējuma cenu un ekipējuma aprakstu.
- **“Darāmo lietu saraksts”** – Apraksta visus lietotāja veidotos uzdevumus, kurus vēlas izpildīt līdz noteiktam datumam. Tās atribūtu kopums sevī ietver aprakstu un datumu.



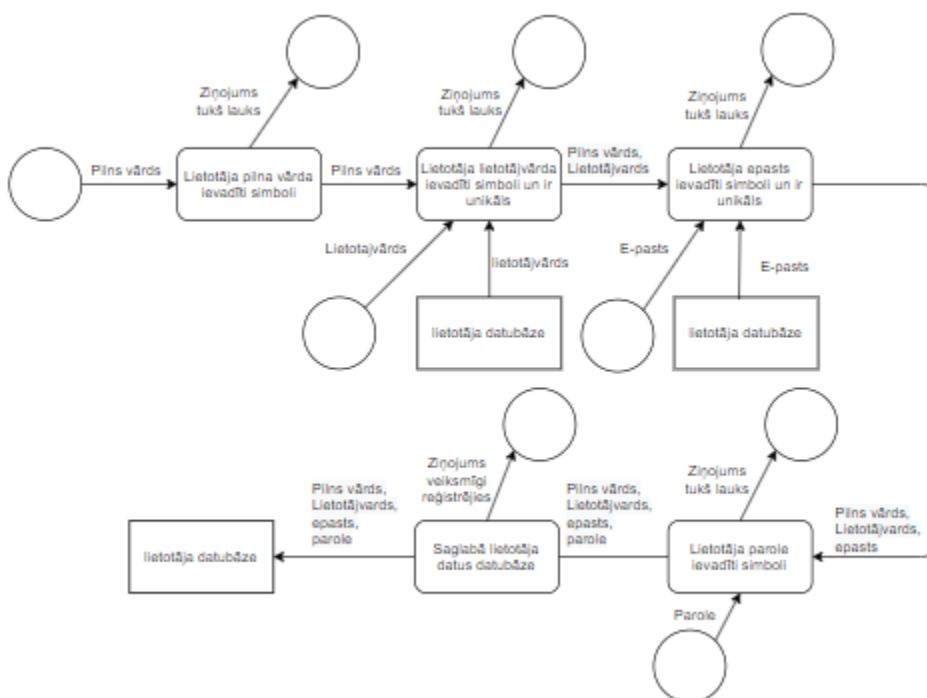
4.2.att. Sistēmas ER diagramma

4.2. Funkcionālais sistēmas modelis

4.2.1. Datu plūsmu modelis

1. Lietotāja reģistrācija

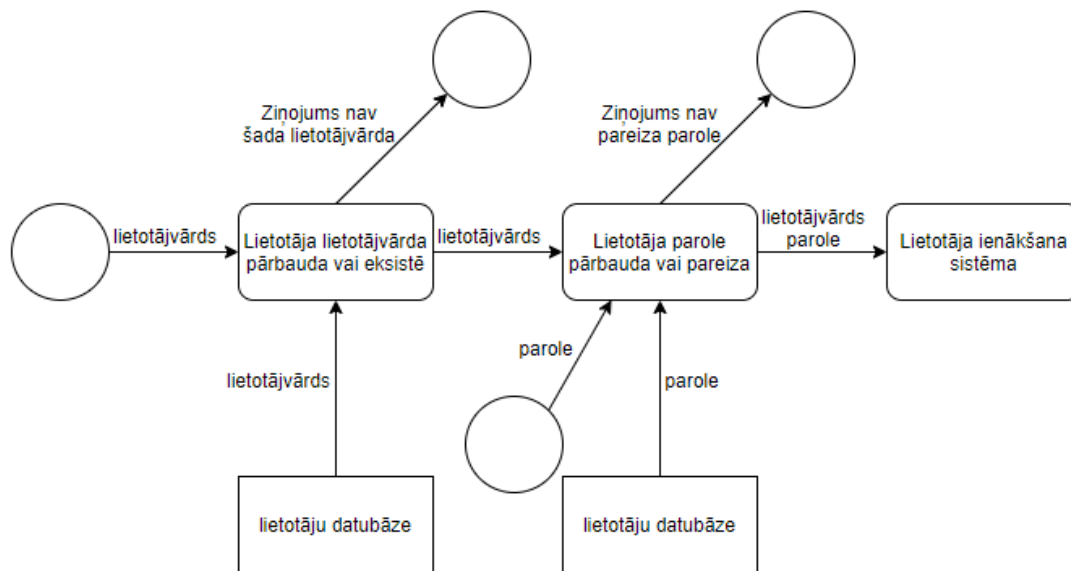
Lietotāja reģistrēšana sistēmā ir pirmais solis, kas jāveic, lai izmantotu aplikāciju. Reģistrācija strādā sekojoši. (skat. 4.3. att.) Lietotājs ievada pilnu vārdu (vārds uzvārds un ja ir tad arī otrs vārds), lietotājvārdu kurš ir unikāls, kuru izmanto, lai pieslēgtos aplikācijai, e-pastu kurš ir unikāls un paroli. Tad sistēma pārbauda vai visi lauki ir aizpildīti un ja nav izvada ziņojumu, ka lauks ir tukšs un tas jāaizpilda. Tad tiek pārbaudīts vai lietotājs ar tādu e-pastu vai lietotājvārdu jau neeksistē, ja nē tad saglabā datus datu bāzē lietotājs, ja eksistē prasa ievadīt citu e-pastu un/vai lietotājvārdu.



4.3. att. Lietotāja reģistrācija

2. Lietotāja ienākšana sistēmā

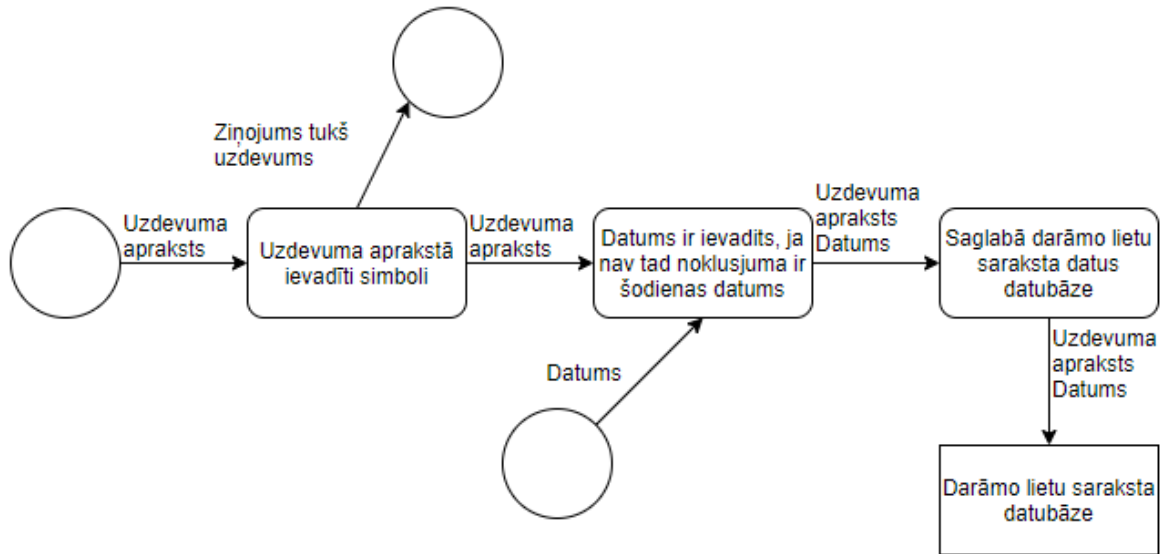
Lai lietotājs varētu ienākt sistēmā pēc reģistrēšanas viņam ir jāpieslēdzas. Pieslēgšanas strādā sekojoši (skat. 4.4. att.). Lietotājs ievada lietotājvārdu un paroli ar kuru reģistrējās, sistēma pārbauda vai ievadītie dati atbilst datiem datu bāzē. Ja dati neatbilst, tad izvada paziņojuma, ka lietotājvārds vai parole nav pareiza. Ja dati ir pareizi lietotājs tiek ielaists aplikācijā.



4.4.att. Lietotāja pieslēgšanās aplikācijai

3. Darāmo lietu ievade

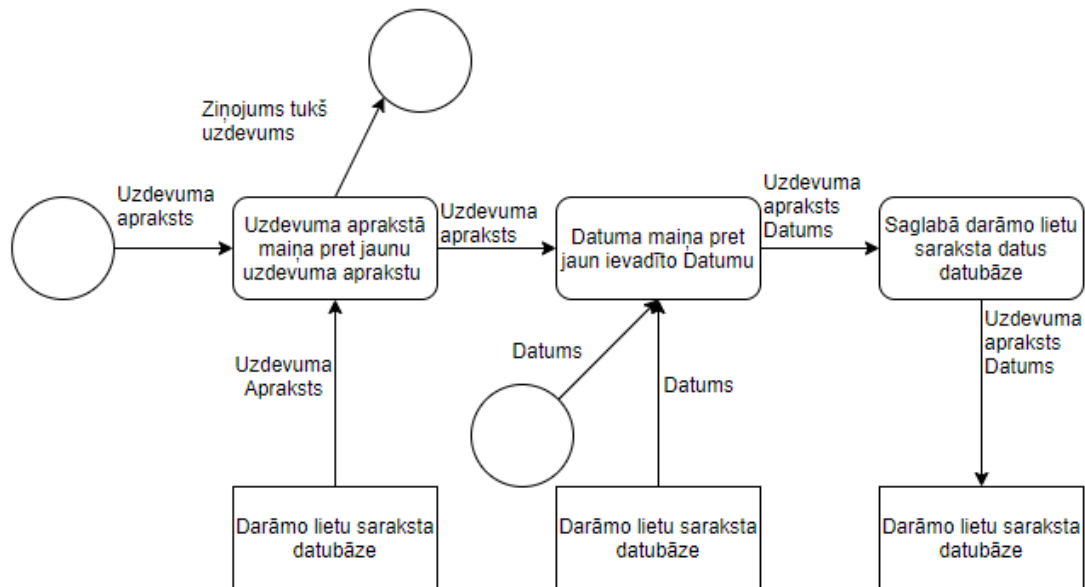
Darāmo lietu saraksta ievade strādā sekojoši (skat. 4.5. att.). Lietotājs ievada uzdevuma aprakstu, ja neievada, tad izmet paziņojumu, ka lauks ir tukšs, pēc tā izvēlas datumu līdz kuram jāizpilda uzdevums, ja lietotājs neizvēlas, tad datubāze tiek saglabāts uzdevums ar šodienas datumu un ievadīto aprakstu.



4.5. att. Darāmo lietu ievade

4. Darāmo lietu rediģēšana

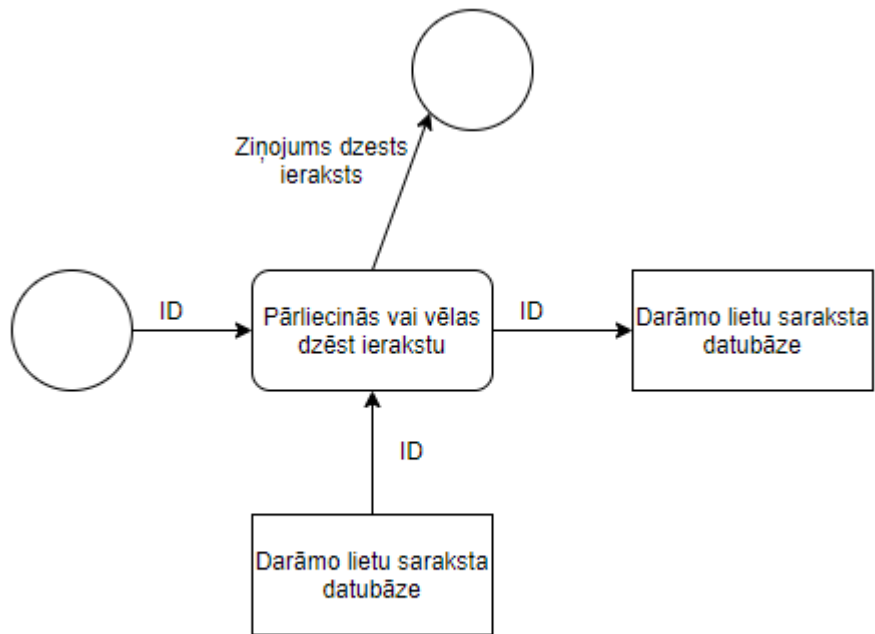
Darāmo lietu saraksta rediģēšana strādā sekojuši (skat. 4.6. att.). Lietotājs izvēlas, kuru uzdevumu vēlas mainīt. Izvēlētajā uzdevuma ieraksta jaunu aprakstu un datumu, ja ne izvēlas jaunu datumu paliek iepriekš ievadītais. Datus saglabā datubāzē.



4.6. att. Darāmo lietu rediģēšana

5. Darāmo lietu dzēšana

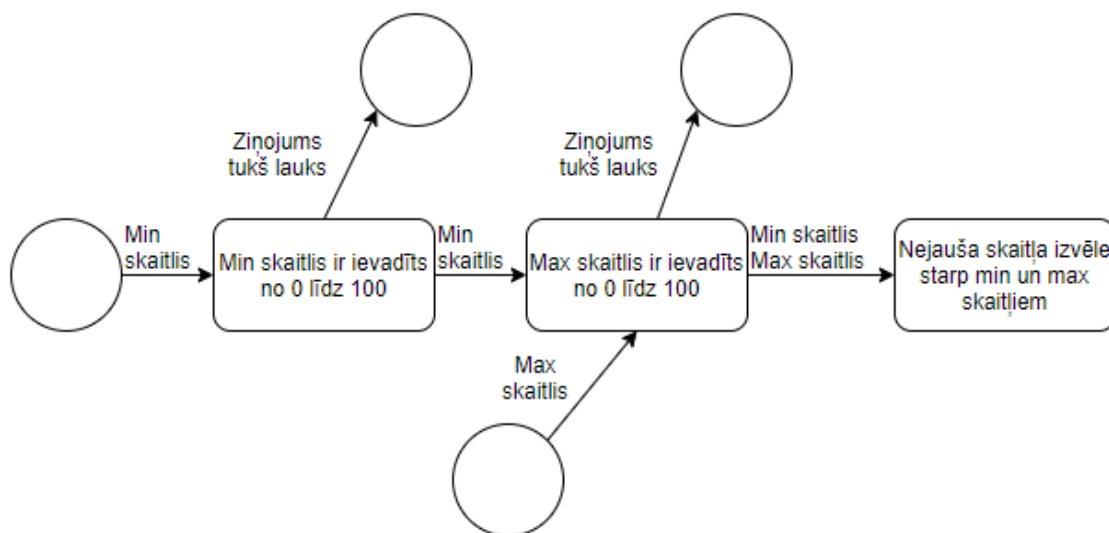
Darāmo lietu dzēšana strādā sekojoši (skat. 4.7. att.). Lietotājs izvēlas ierakstu, kuru dzēst, izvada ziņojumu, lai pārbauda vai tiešam lietotājs vēlas dzēst, ja vēlas dzēst, tad ieraksts tiek dzēsts no datubāzes.



4.7.att Darāmo lietu dzēšana

6. Nejauša skaitļa ģenerēšana

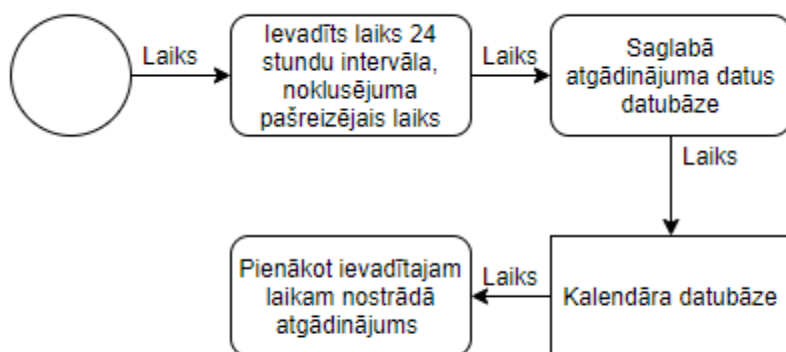
Nejauša skaitļa ģenerēšana strādā sekojoši (skat. 4.8. att.). Lietotājs ievada mazāko iespējamo vingrinājuma izpildes reižu skaitu un maksimālo vingrinājuma izpildes reižu skaitu 0 un 100 intervālā. Ja abi lauki ir aizpildīti, tad aplikācija nejauši izvēlas skaitli dotajā intervālā.



4.8. att. Nejauša skaitļa ģenerēšana

7. Atgādinājuma uzstādīšana

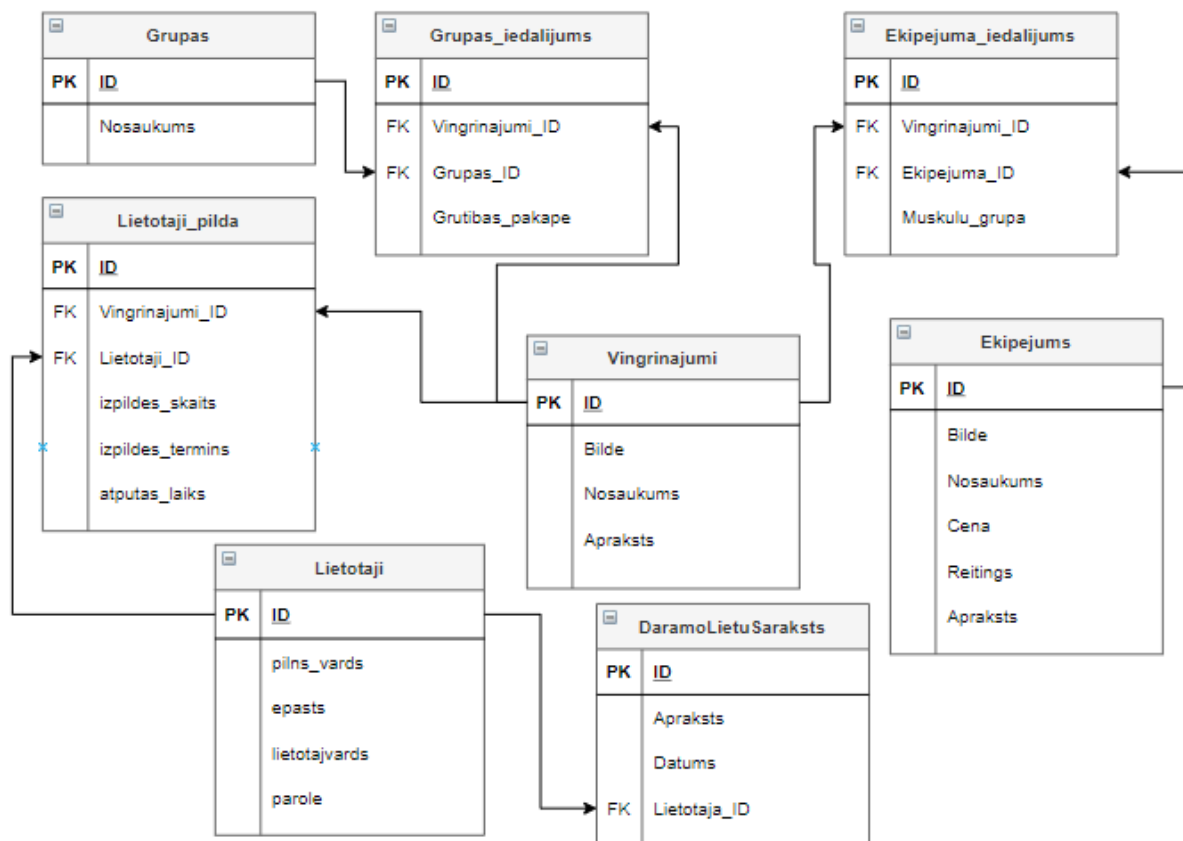
Atgādinājuma uzstādīšana strādā sekojoši (skat. 4.9. att.). Lietotājs ievada laiku, kurā vēlas, lai nostrādā atgādinājums divdesmit četrus stundu intervāla. Ievadīto laiku ievieto kalendāra datubāze. Kad pienāk lietotāja ievadītais laiks nostrādā atgādinājums.



4.9. att. Atgādinājuma uzstādīšana

5. DATU STRUKTŪRAS APRAKSTS

Datubāzē sastāv no 8 tabulām, kas satur informāciju par sistēmas lietotājiem, lietotāja darāmo lietu sarakstu, vingrinājumiem, ekipējumu un grupām. Vingrinājumi ir galvenais platformas objekts, kas savienojas ar vairākām tabulām. Zemāk dotajā attēlā ir parādīta datu bāzes struktūra, kas demonstrē datu kolonu izvietojumu pa tabulām, kā arī tabulu savstarpējo saistību:



5.1.att. Tabulu struktūra

Tabulā “Grupas” glabājas informācija par vingrinājumu grupu nosaukumiem.

5.2.tabula

“Grupas” tabulas struktūra

Nr.	Nosaukums	Tips	Garums	Piezīmes
1	ID	Autonumber	-	Primāra atslēga
2	Nosaukums	Varchar	255	Grupas nosaukums

Tabulas “**Grupas_iedalijums**” ir starp tabula, kas realizē saiti daudzi pret daudziem, savienojot tabulas: “Grupas” un “Vingrinājumi” , jo vienam vingrinājumam varētu būt vairākas grupas. Tabula glabājas vingrinājuma numurs, grupas identifikators un Grūtības pakāpes nosaukums.

5.3.tabula

“Grupas_iedalijums” tabulas struktūra

Nr.	Nosaukums	Tips	Garums	Piezīmes
1	ID	Autonumber	-	Primāra atslēga
2	Vingrinajumi_ID	Int	100	Vingrinājuma numurs
3	Grupas_ID	Int	100	Grupas id
4	Grutibas_pakape	Varchar	255	Grūtības pakāpes nosaukums

Tabulas “**Vingrinajumi**” glabājas informācija par visiem aplikācijas vingrinājumiem, kurus var izpildīt lietotājs. Tabulā glabājas dati par vingrinājuma attēlu, vingrinājuma nosaukumu un vingrinājuma aprakstu.

5.4.tabula

“Vingrinajumi” tabulas struktūra

Nr.	Nosaukums	Tips	Garums	Piezīmes
1	ID	Autonumber	-	Primāra atslēga
2	Bilde	Image	-	Vingrinājuma attēls
3	Nosaukums	Varchar	255	Vingrinājuma nosaukums
4	Apraksts	Text	500	Vingrinājuma apraksts

Tabulas “**Lietotaji_pilda**” ” ir starp tabula, kas realizē saiti daudzi pret daudziem, savienojot tabulas: “Vingrinājumi” un “Lietotaji”, jo vienu vingrinājumu var veikt vairāki lietotāji. Tabula glabājas vingrinājuma numurs, lietotāja identifikators, atkārtojumu skaits ir skaits kurš nosaka cik reizes jāatkārto vingrinājums vienā mēģinājumā, izpildes termiņu var uzstādīt lietotājs, kad viņš vēlas veikt treniņu, kā atgādinājumu un atpūtas laiks ir laiks cik ilgi atpūšas starp vingrinājumiem.

5.5.tabula

“Lietotaji_pilda” tabulas struktūra

Nr.	Nosaukums	Tips	Garums	Piezīmes
1	ID	Autonumber	-	Primāra atslēga
2	Vingrinajumi_ID	Int	100	Vingrinājuma numurs
2	Lietotaji_ID	Int	100	Lietotāja id
3	izpildes_skaits	Int	100	Atkārtojumu skaits
4	izpildes_termins	Date	-	Izpildes datums
5	atputas_laiks	Time	-	Atpūšanas laiks

Tabulas “**Lietotaji**” glabājas visa informāciju par lietotāju. Tabulas atribūtiem “epasts” un “parole” ir jābūt unikāliem. Tabula glabājas lietotāja pilns vārds, lietotāja e-pasts, lietotāja lietotājavārds, kuru lietotājs izmanto, lai ienāktu sistēma kopā ar lietotāja paroli.

5.6.tabula

“Lietotaji” tabulas struktūra

Nr.	Nosaukums	Tips	Garums	Piezīmes
1	ID	Autonumber	-	Primāra atslēga
2	pilns_vards	Text	-	Lietotāja pilns vārds
3	epasts	Varchar	50	Unikāls lietotāja e-pasts
4	lietotajavards	Varchar	50	Unikāls lietotāja lietotājavārds
5	parole	Text	-	Lietotāja parole

Tabulas ” **Ekipējuma_iedalījums**” ” ir starp tabula, kas realizē saiti daudzi pret daudziem, savienojot tabulas: “Vingrinājumi” un “Ekipējums”, jo vienam vingrinājumam varētu būt vairāki ekipējuma veidi. Tabula glabājas vingrinājuma numurs, ekipējuma identifikators un muskuļu grupas nosaukums, kurai atbilst vingrinājums un ekipējums.

5.7.tabula

“Ekipējuma_iedalījums” tabulas struktūra

Nr.	Nosaukums	Tips	Garums	Piezīmes
1	ID	Autonumber	-	Primāra atslēga
2	Vingrinājumi_ID	Int	100	Vingrinājuma numurs
3	Ekipējuma_ID	Int	100	Ekipējuma id
4	Muskulu_grupa	Varchar	255	Muskuļu grupas nosaukums

Tabulas” **Ekipējums**” glabājas visa informāciju par ekipējumu. Tabula ekipējums tiek izvadīta visa informācija lietotājam par esošo ekipējumu, tā cenu, reitings, apraksts, izskata attēlu un nosaukums.

5.8.tabula

“Ekipējums” tabulas struktūra

Nr.	Nosaukums	Tips	Garums	Piezīmes
1	ID	Autonumber	-	Primāra atslēga
2	Bilde	Image	-	Ekipējuma attēls
3	Nosaukums	Varchar	255	Ekipējuma nosaukums
4	Cena	Decimal	100	Ekipējuma cena
5	Reitings	Int	100	Ekipējuma Reitings
6	Apraksts	Text	-	Ekipējuma apraksts

Tabulas “**DarāmoLietuSaraksts**” glabājas visa informāciju par lietotāja darāmiem uzdevumiem ko pats ir izveidojis. Tabula DaramoLietuSaraksts tiek izvadīta visa informācija par lietotāja pievienotajiem uzdevumiem, tā aprakstu un izpildes datumu.

5.9.tabula

“DaramoLietuSaraksts” tabulas struktūra

Nr.	Nosaukums	Tips	Garums	Piezīmes
1	ID	Autonumber	-	Primāra atslēga
2	Apraksts	Varchar	500	Uzdevuma apraksts
3	Datums	Date	100	Uzdevuma izpildes datums
4	Lietotāja_ID	Varchar	255	Lietotāja ID

6. LIETOTĀJA CEĻVEDIS

6.1. Sistēmas prasības aparatūrai un programmatūrai

Lai sāktu veiksmīgu darbu pie sistēmas, ir nepieciešams:

- ❖ Dators ar uzstādītu programmatūra;
- ❖ Android mobila ierīce (Neobligāti).

Aplikāciju var palaist gan uz datora, gan arī uz mobilas ierīces. Mobila aplikācija ir veidota pēc moderniem standartiem, kas krietni samazina iespējamo lietotāju skaitu, bet tas tika darīts, lai lietotājam nogādātu kvalitatīvāku, ērtāku, ātrāku un skaistāku aplikāciju, kurā lietotājam radīs pozitīvas emocijas pret mobilo aplikāciju. Lai palaistu mobilo aplikāciju uz datora ir nepieciešama programmatūra, kas var kompilēt programmas kodu un izveidot virtuālu emulatoru, kas vizualizēs mobilo aplikāciju. Android Studio ir tā programmatūra, kas mums palīdzēs kompilēt kodu, gan arī, ar emulatora palīdzību izveidot virtuālu mobilo telefonu, uz kura tiks uzinstalēta ‘sthenos’ mobila aplikācija priekš lietošanas uz datora. Lai varētu lietot Android Studio un emulatoru ir nepieciešama šāda specifikācija priekš datora (skat. 6.1.1. tabulu).

Minimālās prasības lietotāja programmatūrai un viņa datora specifikācijām

	Microsoft Windows	Mac	Linux
Operētājsistēmas versija	Microsoft Windows 7/8/10 64-bit	Mac® OS X® 10.10 (Yosemite) vai jaunāka versija, līdz 10.14 (macOS Mojave)	GNOME vai KDE darbvirsma
Brīvpiekļuves atmiņa (RAM)	4 GB RAM minimums; Ieteicams 8 GB RAM		
diska vieta	Vismaz 2 GB pieejamās atmiņas, 4 GB Ieteicams (500 MB IDE + 1,5 GB Android SDK un emulatora sistēmas attēlam)		
Minimālā nepieciešamā JDK versija	Java Development Kit 8		
Minimālā ekrāna izšķirtspēja	1280 x 800		
Procesors	vismaz 4 kodolu procesors, neatkarīgi no tā, vai tas ir i5 vai i7, vai i9		

Android emulatoram ir papildu prasības, kas pārsniedz Android Studio pamatsistēmas prasības, kas aprakstītas turpmāk:

- ❖ SDK rīki 26.1.1 vai jaunāka versija;
- ❖ Windows: CPU ar UG (neierobežotu viesu) atbalstu;
- ❖ Intel Hardware Accelerated Execution Manager (HAXM) 6.2.1 vai jaunāka versija (ieteicams HAXM 7.2.0 vai jaunāka versija);
- ❖ Intel procesors ar Intel VT-x, Intel EM64T (Intel 64) un Execute Disable (XD) Bit funkcionalitātes atbalstu vai AMD procesors ar AMD Virtualization (AMD-V) atbalstu.

Android Studio versijai ir jābūt 4.1.3 vai jaunākai. Neskaitot 6.1.1. tabulā aplūkotās prasības, lietotājam ir nepieciešama tastatūra un pele vai skārienjūtīgais paliktnis, lai spētu

ievadīt un apstrādāt informāciju sistēmā. Priekš datora prasībām ir ieteicams lietot ieteicamo specifikāciju nevis minimālo, lai viss strādātu gludāk un ātrāk, jo ar minimālajiem prasībām emulatora darbība ir pasliktināta un var likties ka aplikācija raustās vai reaģē ļoti lēnām uz lietotāja ievadi.

Lai palaistu aplikāciju uz personīgas mobilas ierīces, mobilai ierīcei ir jābūt šādai specifikācijai.

6.1.2. Tabula

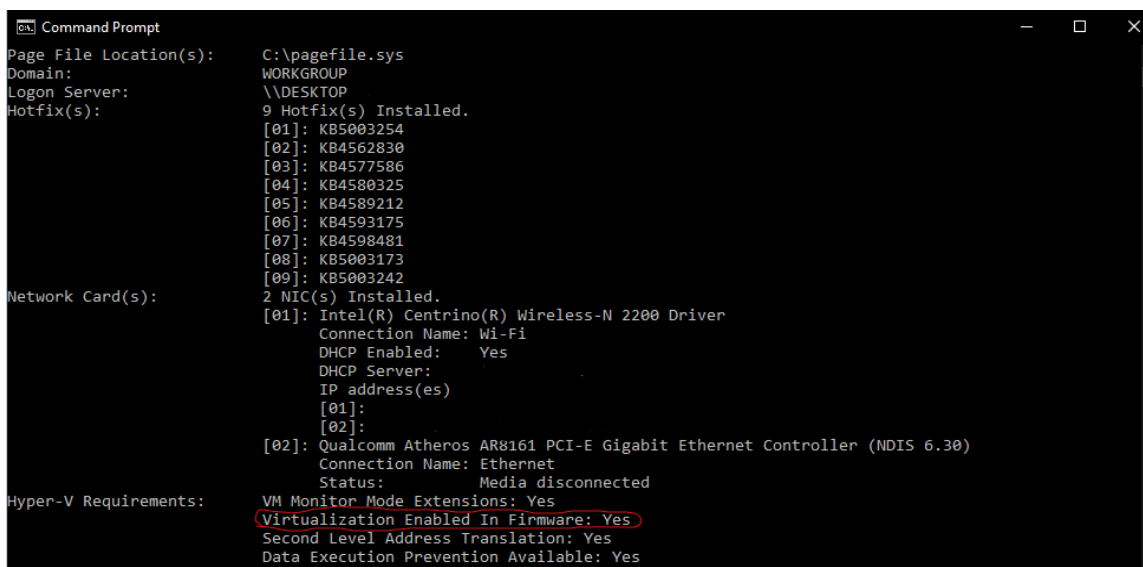
Minimālas un ieteicamas android mobila telefona specifikācija

	Minimums	Ieteicams
Android versija	Android 6.0(API līmenis 23)	Android 9(API līmenis 28) un jaunāks
Procesors	Snapdragon 439	Snapdragon 835 un jaunāks
Brīvpiekļuves atmiņa (RAM)	4GB	8GB
Atmiņas vieta	140MB vai vairāk	

Android versijām, kas ir zem android versijas 6.0, mobilā aplikācija nedarbosies, jo rīki, kas tika izmantoti uz vecajām versijām mūsdienas vairs nav aktuāli, kā arī vecākas versijas nav spējīgas atbalstīt jaunākas funkcijas, kas tiek izmantoti jaunākajās android versijās. Vislabāk ir izmantot vienu no jaunākajām android versijām, jo uz tām aplikācija darbosies, kā paredzēts un nebūs nekādu problēmu ar funkcijām, kas tika taisītas tieši priekš android 11 versijas funkcijām. Atmiņas vietai ir nepieciešams vismaz 140 MB, jo aplikācija aizņem tādu lielumu no atmiņas. Tāpat kā ar Android Studio ir ieteicams 8GB brīvpiekļuves atmiņas, lai aplikācija neraustītos un strādātu gludi. Jābūt arī uzinstalētai XAMPP programmatūrai, lai tiktu galā ar datu bāzes pusi, kas ietilpst datora minimālajās prasībās (skat. 6.1.1 tabulu).

6.2. Sistēmas instalācija un palaišana

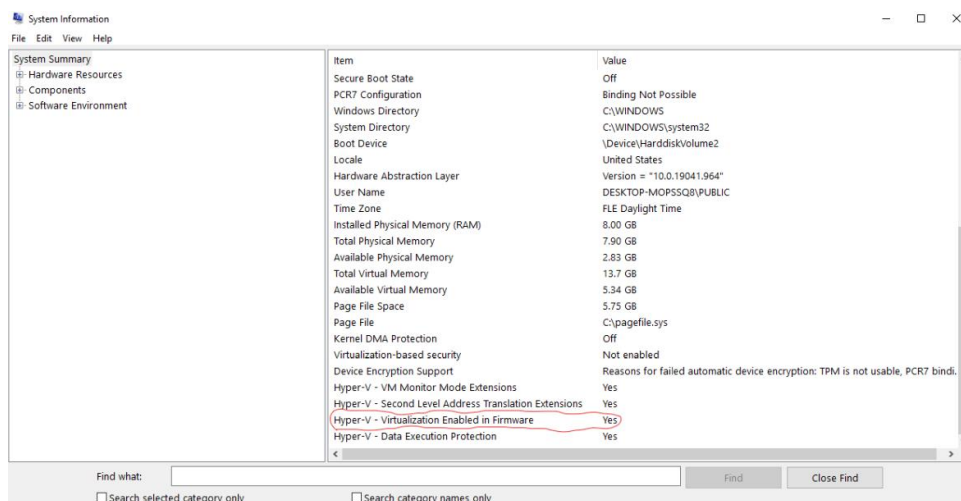
Sistēmai ir divu veidu palaišana. Pirmā no tām ir palaišana uz datora. Lai palaistu aplikāciju uz windows 10 datora caur emulatoru no sākuma ir jāpārbauda vai programmaparatūrā iespējota virtualizācija, ko var apskatīties izmantojot konsoli vai run funkciju. Ja izmanto konsoles pieeju, tad ierakstot windows 10 meklēšanas josla ‘Command prompt’ vai ‘CMD’, palaižot konsoli komand rindā ierakstiet ‘systeminfo.exe’, kur jums izvadīs visu informāciju par jūsu sistēmu (skat. 6.2.1. att.).



```
Command Prompt
Page File Location(s): C:\pagefile.sys
Domain: WORKGROUP
Logon Server: \\DESKTOP
Hotfix(s): 9 Hotfix(s) Installed.
[01]: KB5003254
[02]: KB4562830
[03]: KB4577586
[04]: KB4580325
[05]: KB4589212
[06]: KB4593175
[07]: KB4598481
[08]: KB5003173
[09]: KB5003242
Network Card(s): 2 NIC(s) Installed.
[01]: Intel(R) Centrino(R) Wireless-N 2200 Driver
Connection Name: Wi-Fi
DHCP Enabled: Yes
DHCP Server:
IP address(es)
[01]:
[02]:
[02]: Qualcomm Atheros AR8161 PCI-E Gigabit Ethernet Controller (NDIS 6.30)
Connection Name: Ethernet
Status: Media disconnected
Hyper-V Requirements: VM Monitor Mode Extensions: Yes
Virtualization Enabled In Firmware: Yes
Second Level Address Translation: Yes
Data Execution Prevention Available: Yes
```

6.2.1.att. Command prompts pēc ‘systeminfo.exe’ komandas izpildes

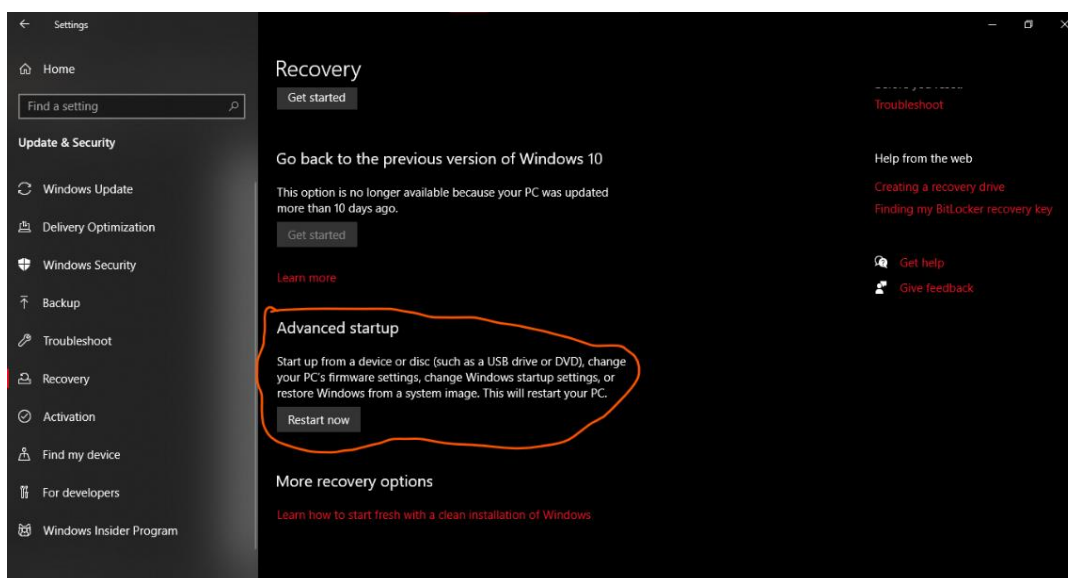
Ja izmantojat run pieeju, tad ierakstot windows 10 meklēšanas josla ‘run’ jums parādīsies mazs logs kurā jums jāieraksta ‘msinfo32’, kur jums arī izvadīs visu informāciju par jūsu sistēmu, bet tā būs vizuāli (skat. 6.2.2. att.).



6.2.2.att. Sistēmas info pēc komandas ‘msinfo32’ ievades run loga

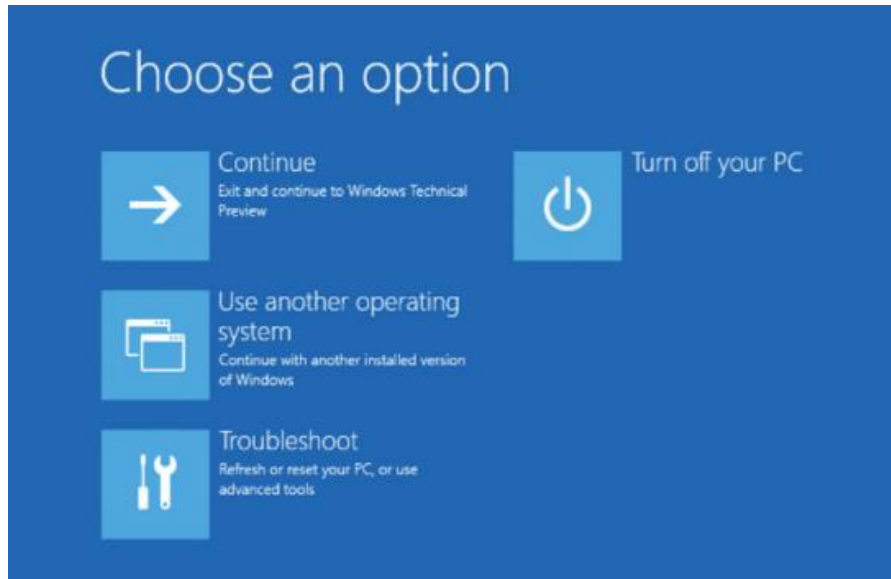
Tālāk atrodi sadaļu ‘Hyper-V Requirements’ atrodi ‘Virtualization Enabled In Firmware’, ja blakus ir vārds ‘Yes’, tad viss ir kārtība un tālāk nav jāveic nekādas darbības, taču ja blakus ‘Virtualization Enabled In Firmware’ ir ‘No’, tad būs jāveic tālāk norādītas darbības.

Meklēšanas josla ierakstiet ‘Settings’, atverot iestatījumus atrodi un atveriet sadaļu ‘Update & Security’. Kreisajā sānā ir navigācijas josla, kurā jums ir jāatrod un jāatver sadaļa ‘Recovery’. Atverot šo sadaļu nospiediet uz pogas ‘Restart now’ zem virsraksta ‘Advanced startup’ (skat. 6.2.3. att.).



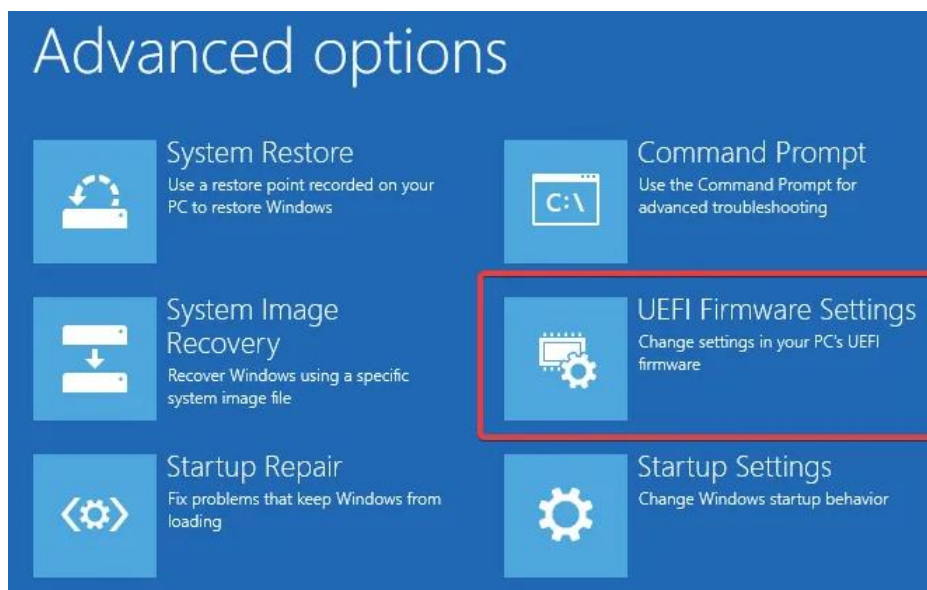
6.2.3. att. Windows 10 Settings – Update & Security – Recovery – Advanced startup sadaļa

Datoram restartējoties parādīsies zils ekrāns ar izvēles opcijām (skat. 6.2.4. att.).



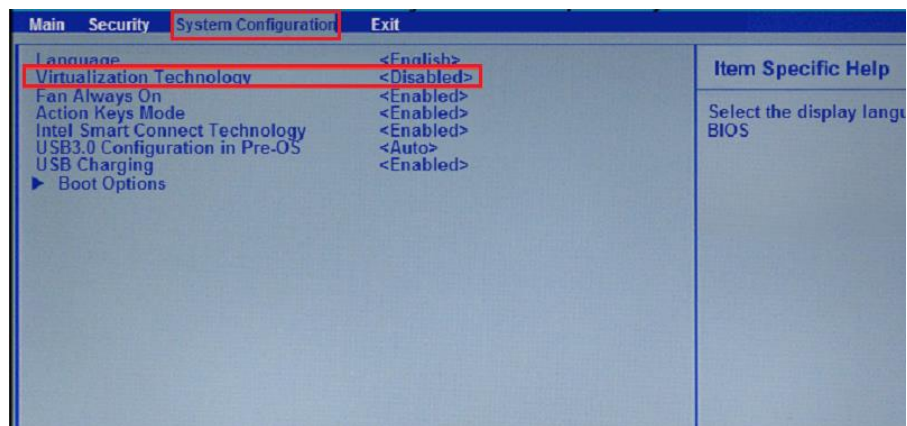
6.2.4. att. Advanced startup logs

Jāizvēlas ir ‘Troubleshoot’ opcija, zem kuras parādīsies nākamais logs, kur ir jāizvēlas no divām opcijām ‘Advanced options’. Atverot pielāgotas opcijas ir jāizvēlas sadaļā ‘UEFI Firmware Settings’, kur jums prasīs restartēt (skat. 6.2.5. att.).



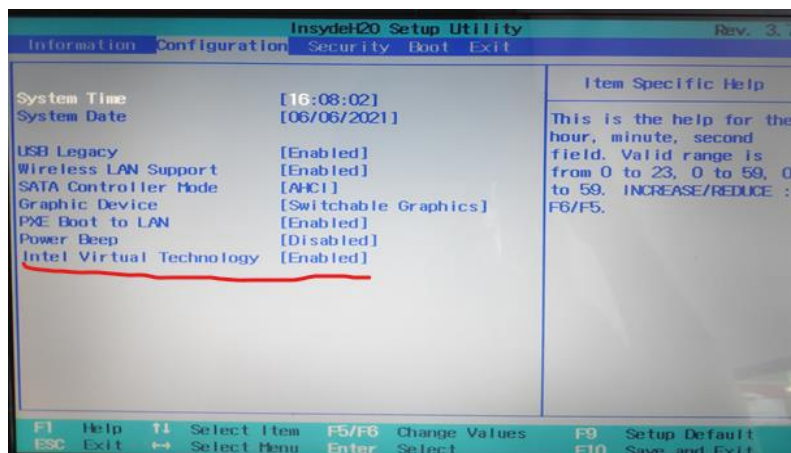
6.2.5. att. Pielāgotas opcijas

Pēc restarta būs melns ekrāns ar izvēlas opcijām. Jums ir jāizvēlas ‘BIOS setup’. Kad jums atvērās ‘BIOS Settings’ izmantojot bultiņas uz tastatūras pārvietojaties uz sadaļu ‘System Configuration’ un zem tās izvēlaties ‘Virtualization Technology’ ar ‘Enter’ pogu, jums parādīsies izvēlnē ar divām opcijām ‘Enabled’ un ‘Disabled’, izvēlaties ‘Enabled’ un nospiediet uz tastatūras pogu ‘F10’, lai saglabātu un izietu (skat. 6.2.6. att.).



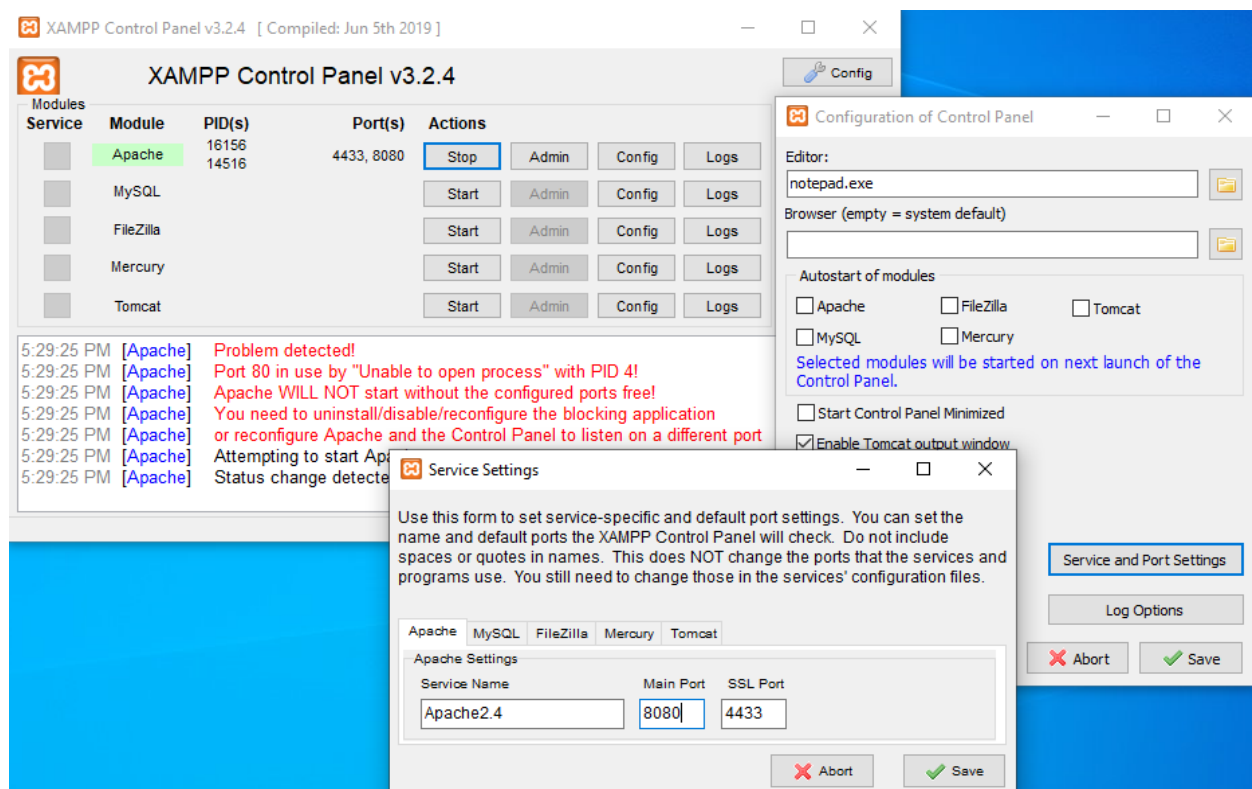
6.2.6. att. BIOS settings, system configuration, virtualizācija ir atslēgta

Ja jums nav ‘UEFI Firmware Settings’ opcijas, tad restartējiet datoru, atkārtoti spiežot pogu ‘F2’, kas atrodas uz tastatūras. Jums atvērsies ‘InsydeH20 Setup Utility’. Izmantojot bultiņas uz tastatūras pārvietojaties uz ‘Configuration’ sadaļu un zem tās izvēlaties ‘Virtualization Technology’ ar ‘Enter’ pogu, manā gadījumā tas ir ‘Intel Virtual Technology’, jums parādīsies izvēlnē ar divām opcijām ‘Enabled’ un ‘Disabled’, izvēlaties ‘Enabled’ un nospiediet uz tastatūras pogu ‘F10’, lai saglabātu un izietu (skat. 6.2.7. att.).



6.2.7. att. InsydeH20 Setup Utility

Pēc virtualizācijas iespējošanas ir jāpieslēdzas Apache un MySQL ar XAMPP palīdzību. Pirms pieslēgšanās ir jāparliecinās ka XAMPP mapē, 'htdocs' apakšmapē ir ievietota mape 'Sthenos' ar četriem PHP failiem, kas nodrošina komunikāciju starp aplikāciju un datubāzi. Ja mape 'Sthenos' atrodas apakšmapē 'htdocs', tad var savienoties ar Apache un MySQL caur XAMPP Control. Ja savienošanas laikā izmet kļūdu, ka ports ir aizņemts, tad tos var nomainīt 'config' opcija, zem kuras ir jāizvēlas 'Service and Port Settings'. Apache noklusijama galvenais ports ir 80 un SSL ports ir 443, ja kāds no šiem portiem ir aizņemts, tad standarta maina portu 80 uz portu 8080 un 443 portu uz 4433 portu (skat. 6.2.8. att.).



6.2.8. att. XAMPP Apache porta maiņa

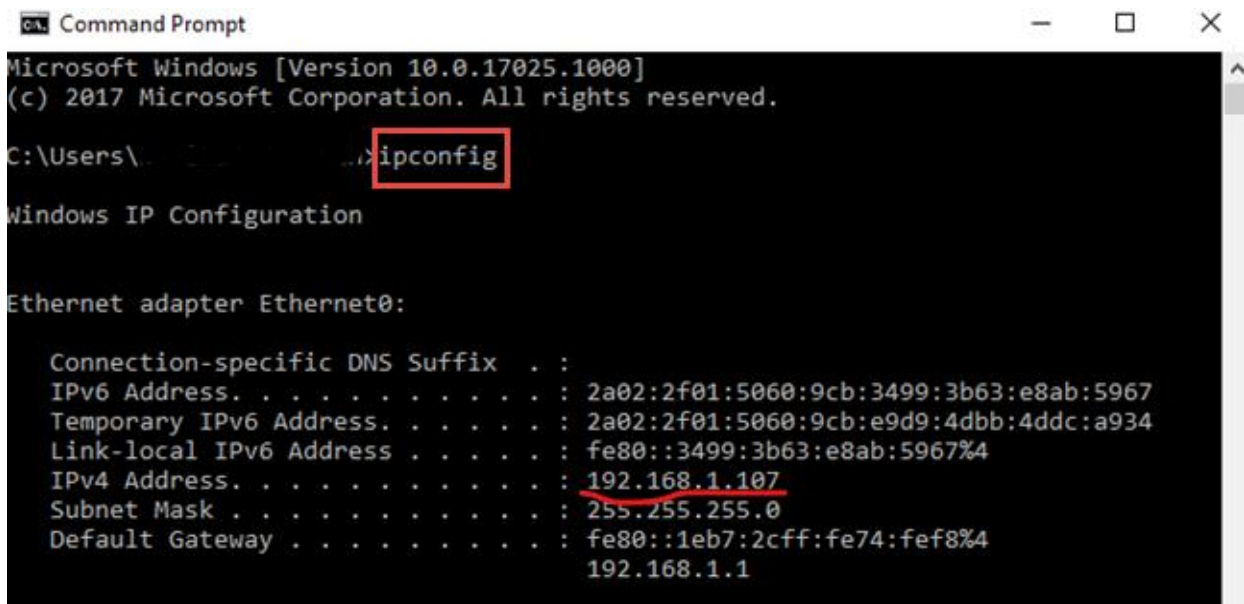
Kad ir izveidots savienojums ar Apache un MySQL ir jādodas uz phpMyAdmin un jāizveido datubāze 'sthenos', kurā ir tabula 'users' ar pieciem atribūtiem (skat. 5.6. tabulu) uz 'localhost' servera, kas ir lokāls serveris, kas izmanto datora IPv4 adresi. XAMPP kontrol panelī blakus MySQL jānospiež poga Admin, kas aizvedīs uz phpMyAdmin serveri tīmekļa vidē.

Pēc tam Android Studio vidē jarediģe kods, Login.java un SignUp.java failos, kur 'IPv4adress' (skat. 6.2.9. att.) vieta jāieraksta sava IPv4 adrese.

```
PutData putData = new PutData( url: "http://IPv4Address:8080/SthenosPHP/login.php", method: "POST", field, data);
```

6.2.9. att. Login. Java fails

Lai atrastu savu IPv4 adresi, konsole jāieraksta komanda 'ipconfig' un izvadīs datus par konfigurāciju, kur jāmekle 'IPv4 Address'(skat. 6.2.10. att.).



```
Microsoft Windows [Version 10.0.17025.1000]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\...>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . : 
    IPv6 Address. . . . . : 2a02:2f01:5060:9cb:3499:3b63:e8ab:5967
    Temporary IPv6 Address. . . . . : 2a02:2f01:5060:9cb:e9d9:4dbb:4ddc:a934
    Link-local IPv6 Address . . . . . : fe80::3499:3b63:e8ab:5967%4
    IPv4 Address. . . . . : 192.168.1.107
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : fe80::1eb7:2cff:fe74:fef8%4
                                192.168.1.1
```

6.2.10. att. Konsoles panelis ar ipconfig

Lai palaistu aplikāciju uz savas mobilas ierīces tā ar vadu ir jāpieslēdz pie datora. Tad jāatver telefona iestatījumus, jāatver informācija par telefonu un atkārtoti jāspiež uz modificētā Android ROM versijas, atkarīgs no ierīces ražotāja, līdz tiek iegūti izstrādātajā tiesības. Kad ir iegūtas izstrādātajā tiesības, jāatver izstrādātajā opcijas un jāiespējo 'USB debugging'. Pēc tā jāatver Android Studio zem emulatorā mobilo ierīču opcijām ir jāizvēlas sava mobilā ierīce un jāpalaiž projekts. Aplikācija tiks lejupielādētā uz jūsu mobilas ierīces priekš lietošanas un automātiski palaista pēc lejupielādes.

6.3. Programmas apraksts

‘Splash’ ekrāns

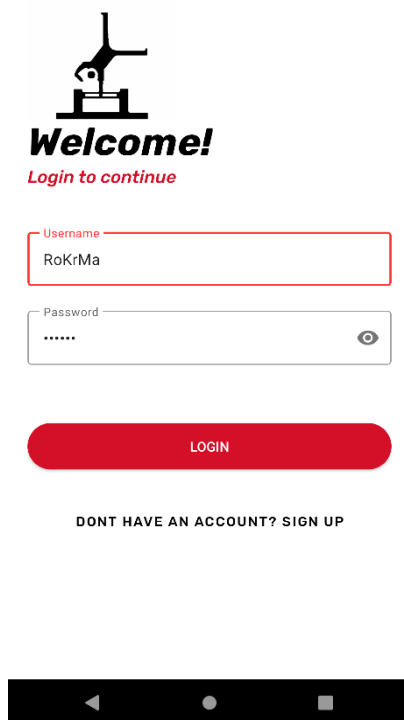
Lietotājam palaižot aplikāciju parādās animēts ‘Splash screen’(skat. 6.3.1. att.), kur no augšas uz apakšu nobrauc aplikācijas logo un no apakšas uz augšu uzbrauc aplikācijas nosaukums uz sarkana fona, kas ilgst piecas sekundes. Mūsdienu aplikācijām šādi ‘Splash screen’ ir ļoti raksturīgi, lai gan nav nozīmīgi aplikācijas darbībai. Šādu ekrānu pievienoju, lai tuvāk atbilstu mūsdienu aplikāciju standartiem.



6.3.1. att. ‘Splash’ ekrāns

Pieslēgšanās ekrāns

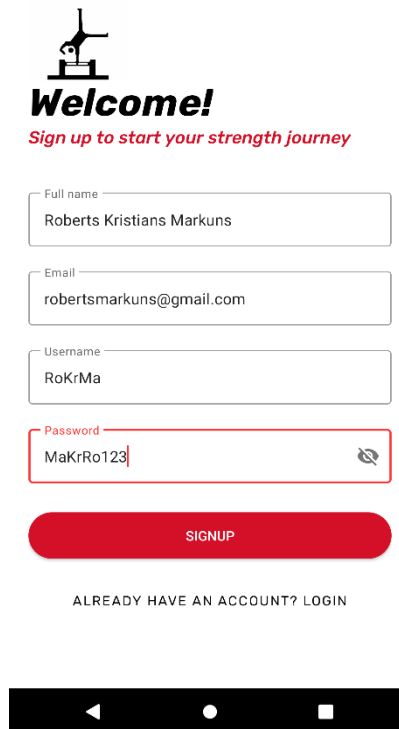
Pēc īsa piecu sekunžu ‘Splash’ ekrāna, lietotājs tiek automātiski nogādāts uz pieslēgšanas ekrānu (skat. 6.3.2. att.). Ekrāna augša ir aplikācijas logo ar tekstu. Zem tā atrodas lietotājvārda un paroles ievades lauki uz kuriem uzspiežot ierakstītais teksts pāriet uz robežlīnijas, lai lietotājs ievadot datus var redzēt kāda lauka viņš tos vada, kā arī paroles lauks ir apslēpts, taču nospiežot uz actiņas varēs redzēt ievadītos simbolus. Zem Ievad laukiem ir ‘LOGIN’ poga, kas salīdzina ievadītos datus ar tiem, kas ir datubāze un kamēr tie tiek pārbaudīti uz ekrāna parādās apla progressa josla, kas griežas apkārt kamēr dati līdz atbildes saņemšanai, ja dati ir pareizi lietotājs tiek ielaists sistēma, ja nē tiek izvadīts paziņojums, ka lietotājvārds vai parole nav pareiza vai lietotājs neeksistē. Zem pieslēgšanas pogas ir poga lietotājiem, kuriem vēl nav konts un vēlas tādu izveidot, nospiežot pogu lietotājs tiek pārvests uz reģistrācijas ekrānu ar slidenu animāciju kur ievad lauki tiek pārbīdīti un parādās divi jauni lauki.



6.3.2. att. Pieslēgšanās ekrāns

Reģistrācijas ekrāns

Lai lietotājs var sākt lietot aplikāciju viņam/-i ir vajadzīgs profils ar kuru varēs pieslēgties. Reģistrācijas ekrāna (skat. 6.3.3. att.) ir četri ievad lauki – pilns vārds, kas ir vārds, uzvārds un otrais vārds ja tāds ir vai vairāk, e-pasts, lietotājvārds un parole- uz kuriem uzspiežot ierakstītais teksts pāriet uz robežlīnijas, lai lietotājs ievadot datus var redzēt kāda lauka viņš tos vada un paroles lauks ir apslēpts, taču nospiežot uz actiņas varēs redzēt ievadītos simbolus. Nospiežot pogu ‘SIGNUP’ aplikācija pārbauda vai visi lauki ir ievadīti un vai e-pasta un lietotājvārda lauks jau nesakrīt ar kādu eksistējošu lietotāju. Zem reģistrācijas pogas ir poga lietotājiem, kuriem jau ir konts un vēlas pieslēgties, nospiežot pogu lietotājs tiek pārvests uz pieslēgšanas ekrānu ar slidenu animāciju kur ievad lauki tiek pārbīdīti un pazūd e-pasta un pilna vārda lauks.

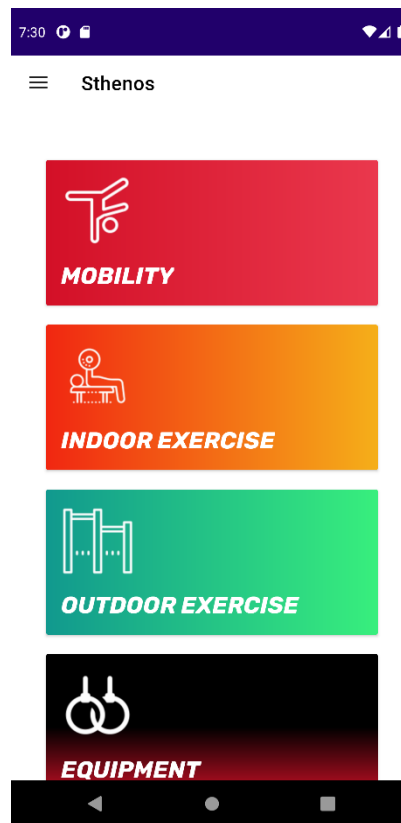


The image shows a registration screen with a black header containing a logo of a person on a bench and the text 'Welcome!' in bold, followed by the tagline 'Sign up to start your strength journey' in red. Below the header are four input fields: 'Full name' with the value 'Roberts Kristians Markuns', 'Email' with 'robertsmarkuns@gmail.com', 'Username' with 'RoKrMa', and 'Password' with 'MaKrRo123'. The password field has a red border and a toggle icon. A red 'SIGNUP' button is below the fields. At the bottom, there is a link 'ALREADY HAVE AN ACCOUNT? LOGIN' and a black navigation bar with three icons: a back arrow, a circle, and a square.

6.3.3. att. pieslēgšanās ekrāns

Mājas ekrāns

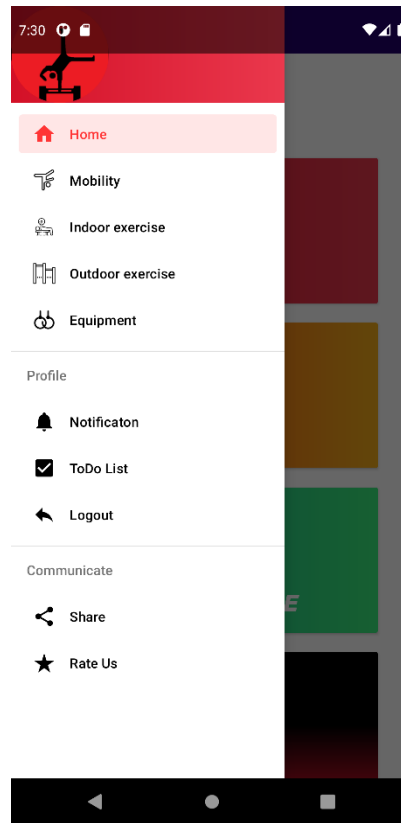
Mājas ekrāns jeb sākuma ekrāna (skat. 6.3.4. att.) ir tas ekrāns ko lietotājs redz pēc pieslēgšanas. Uz tā ir četri vertikāli bloki uz kura nospiežot aizvedīs uz attiecīgo aktivitāti aplikācija. Ja tiks nospiests uz mobilitātes bloka tiks aizvests uz mobilitātes vingrinājuma grupas izvēli, ja uz iekšdurvju bloka, tad tiks aizvests uz iekšdurvju vingrinājuma grupas izvēli un tas pats ar ārdurvju, bet ja uz ekipējuma bloka, tad aizvedīs uz aktivitāti, kur būs vertikāli bloki ar ekipējumu. Augšējā kreisajā stūrī ir navigācijas josla.



6.3.4. att. Sākuma lapa

Sākuma lapas navigācija josla

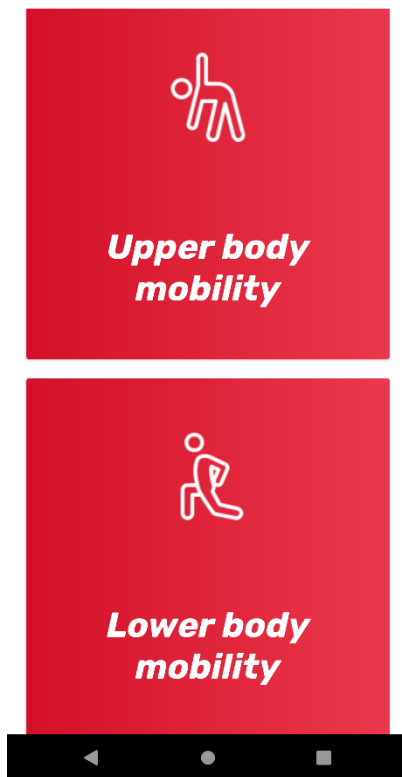
Sākuma lapas navigācija josla, kas atrodas augšējā kreisajā stūrī (skat. 6.3.5. att.). Pirmās piecas izvēles ir sākuma lapa un vingrinājumu vai ekipējuma izvēlē, kas atrodas sākuma lapā. Profila sadaļa atrodas darāmo lietu saraksts, kas pārnes lietotāju uz darāmo lietu saraksta aktivitāti, atgādinājumi, kas pārnes lietotāju uz atgādinājuma aktivitāti un izrakstīšanas poga, kas pārnesīs lietotāju atpakaļ uz pieslēgšanas ekrānu. Komunikācijas sadaļa ir kopīgo un novērtē mūs, kurus izvēloties izveda ziņojumus.



6.3.5. att. Sākuma lapas navigācija josla

Mobilitātes vingrinājuma grupas izvēle

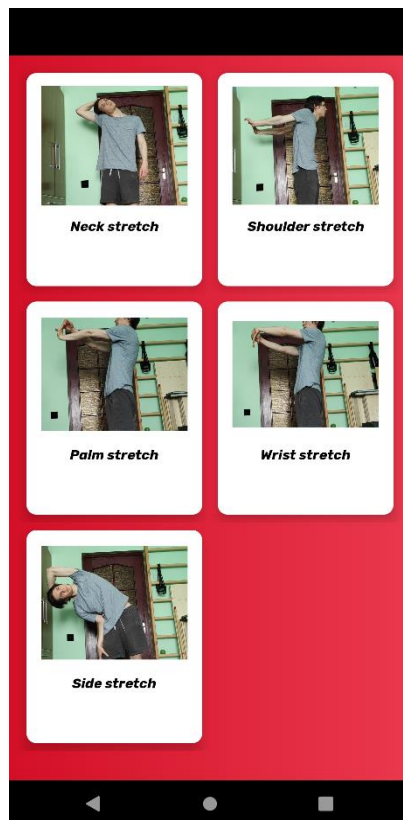
Šajā izvēlnē jāizvēlas kādu ķermeņa daļu vēlas izstaipīt vai augšu vai apakšu. Izvēloties vienu no grupām tiks parādīti vingrinājumi saistīti ar šo grupu. Mobilitātes vingrinājuma grupas izvēle izskatās sekojoši (skat. 6.3.6. att.).



6.3.6. att. Mobilitātes vingrinājuma grupas izvēle

Ķermeņa augšdaļas kustīguma vingrinājuma izvēle

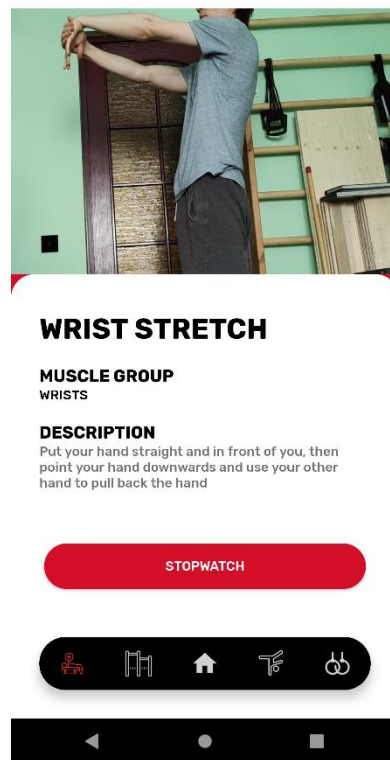
Šajā sadaļā ir jāizvēlas vingrinājums, kuru lietotājs vēlas veikt atkarība no tā, ko vēlas staipīt, tas var būt gan kakls, rokas, sāni, prese u.c. Katram vingrinājumam ir dots vingrinājuma attēls un vingrinājuma nosaukums. Nospiežot uz izvēlēta vingrinājuma atvērsies jauna aktivitāte, kas paskaidros vingrinājumu detalizētāk. Ķermeņa augšdaļas kustīguma vingrinājuma izvēle izskatās sekojoši (skat. 6.3.7. att.).



6.3.7. att. Ķermeņa augšdaļas kustīguma vingrinājuma izvēle

Kustīguma vingrinājumi

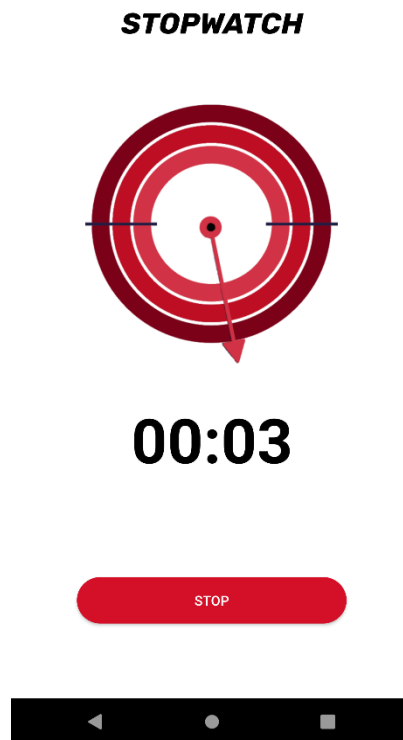
Atverot kādu no vingrinājumiem kustīguma sadaļa, lietotājam tiek parādīta sekojoša informācija par izvēlēto vingrinājumu – vingrinājuma attēls, vingrinājuma nosaukums, vingrinājuma muskuļu grupa un vingrinājuma detalizēts apraksts – kas lietotājam palīdzēs izpildīt vingrinājumu. Vienmēr staipoties vingrinājums tiek veikts konkrētu sekunžu skaitu un koncentrēties uz elpošanu un laika skaitīšanu galvā ir grūti tādēļ ir iebūvēts hronometrs, kuru var izmantot nospiežot uz pogas ‘STOPWATCH’, kas aizvedīs uz hronometra aktivitāti. Kustīguma vingrinājuma aktivitāte izskatās sekojoši (skat. 6.3.8. att.).



6.3.8. att. Kustīguma vingrinājums

Vingrinājuma hronometrs

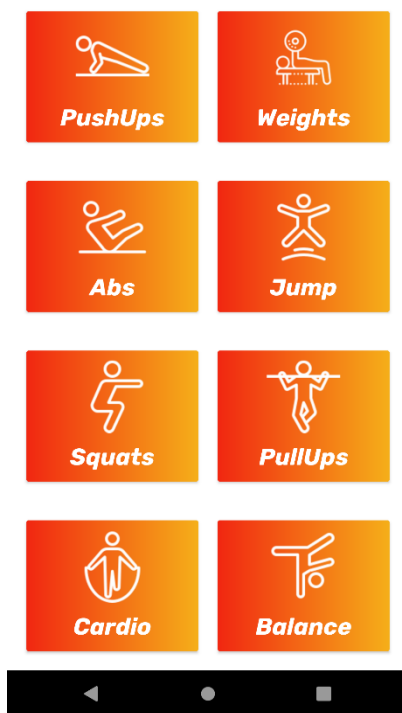
Atverot hronometra aktivitāti ir pulkstenis zem tā ir laika uzskaitītājs ar startēšanas pogu. Nospiežot start pogu, sāk griezties pulkstenis, kurš ir animēts, zem tā skaita laiku, sekundēs, un tiek noslēpta poga start un zemāk parādās apstāšanas poga. Nospiežot apstāšanās pogu tiek apstādināta pulksteņa animācija, laika uzskaite, pazūd stop poga un atkal parādās start pogu. Hronometra aktivitāte izskatās sekojoši (skat. 6.3.9. att.).



6.3.9. att. Hronometra aktivitāte

Iekšdurvju vingrinājuma grupas izvēle

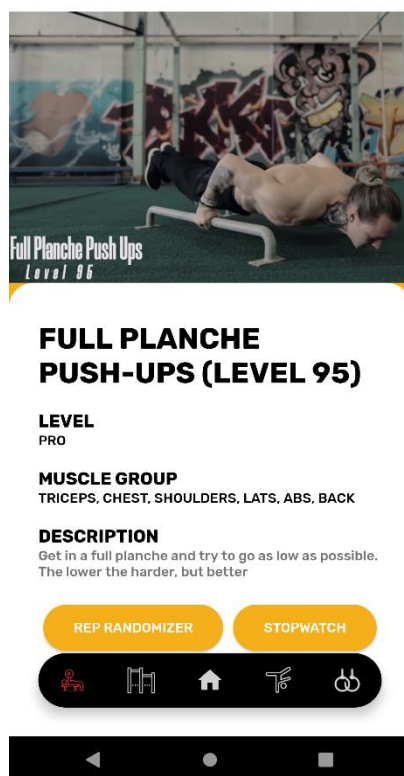
Atgriežoties uz sākuma ekrānu un izvēloties iekšdurvju vingrinājuma grupu, kas atšķiras no kustību grupas ar grupu daudzumu un grupas krāsu. Ārdurvju vingrinājumu grupa atšķiras tikai ar krāsu no iekšdurvju vingrinājumu grupām, iekšdurvju sadaļa ir oranža, bet ārdurvju ir zaļā. Iekšdurvju vingrinājuma grupas izvēles aktivitāte izskatās sekojoši (skat. 6.3.10. att.).



6.3.10. att. Iekšdurvju vingrinājuma grupas izvēle

Iekšdurvju vingrinājumi

Atverot kādu vingrinājumu iekšdurvju sadaļā, lietotājam tiek izvadīta tāda paša veida informācija ka kustību sadaļā: vingrinājuma attēls, vingrinājuma nosaukums, vingrinājuma muskuļu grupa un vingrinājuma detalizēts apraksts. Papildus tam nāk arī vingrinājuma grūtības līmenis. Kā arī papildus hronometram nāk vingrinājumu izpildes skaita ģenerētājs. Iekšdurvju vingrinājumu aktivitāte izskatās sekojoši (skat. 6.3.11. att.).



6.3.11. att. Iekšdurvju vingrinājumi

Vingrinājuma izpildes skaita ģenerētājs

Atverot vingrinājuma izpildes skaita ģenerētāju ir lauks priekš izvades, divi ievades lauki minimālais izpildes reižu skaits un maksimālais izpildes reižu skaits zem tā ir poga, kas ļauj ģenerēt skaitli. Vingrinājuma izpildes skaita ģenerētāja aktivitāte izskatās sekojoši (skat. 6.3.12. att.).

REP RANGE RANDOMIZER

99

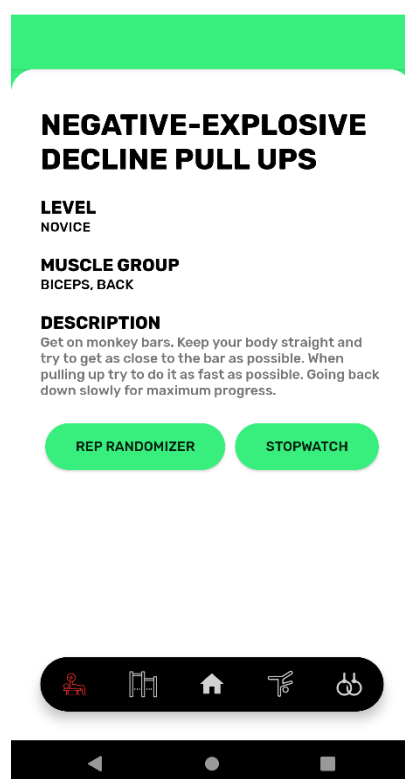
Min	Max
<input type="text" value="0"/>	<input type="text" value="100"/>
<div>ROLL</div>	



6.3.12. att. Vingrinājuma izpildes skaita ģenerētājs

Peldošā navigācijas josla

Peldošā navigācijas josla (skat. 6.3.13. att.) atrodas ekipējuma, vingrinājuma, atgādinājuma un darāmo lietu saraksta aktivitāšu apakša un tas ir piesprausts pie ekrāna, kas nozīme pat, ja lietotājs patīs vingrinājumu vai ekipējumā aktivitāti uz leju vai augšu tā atrašanas uz vieta nemainīsies, bet gan sekos līdzi. Uz navigācijas joslas var ātri pārvietoties - iekšdurvju sadaļu, ārdurvju sadaļu, sākuma ekrānu, kustību sadaļu un ekipējuma sadaļu – tādā secībā. Uzspiežot uz kādas no sadaļām tiek izvadīts sadaļas nosaukums nomaina tās ikonas krāsu uz sarkanu un pāriet uz izvēlēto sadaļu.



6.3.13. att. Peldošā navigācijas josla

Ekipējuma izvēle

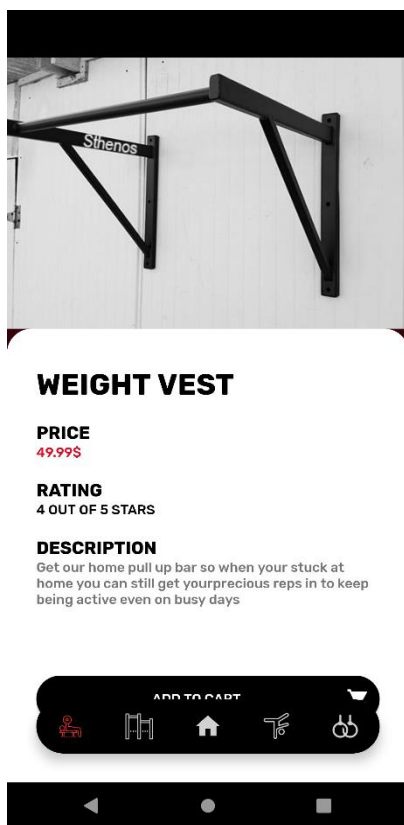
Ekipējuma izvēle ir vertikāli bloki ar ekipējuma attēlu un ekipējuma nosaukumu. Nospiežot uz konkrēta vingrinājumā tiks atvērta jauna aktivitāte ar sīkāku informāciju par ekipējumu. Ekipējuma izvēles aktivitāte izskatās sekojoši (skat. 6.3.14. att.).



6.3.14. att. Ekipējuma izvēle

Ekipējums

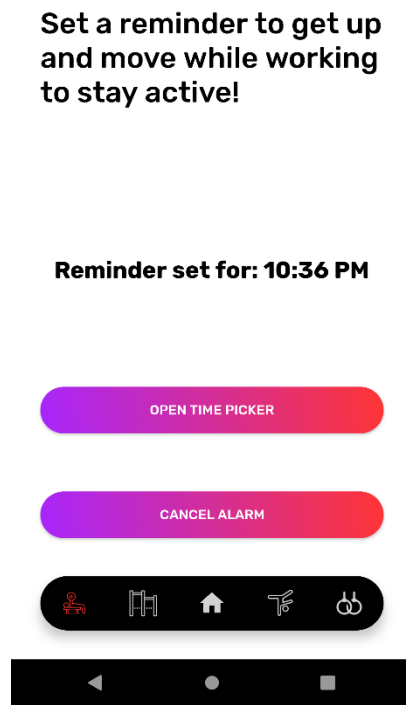
Ekipējuma logā tiek izvadīta sīkāka informācija par ekipējumu: ekipējuma attēls, ekipējuma nosaukums, ekipējumā cena, ekipējuma reitings, ekipējuma apraksts. Zem informācijas ir poga, kura nospiežot izveda paziņojumu. Ekipējuma aktivitāte izskatās sekojoši (skat. 6.3.15. att.).



6.3.15. att. Ekipējums

Atgādinājums

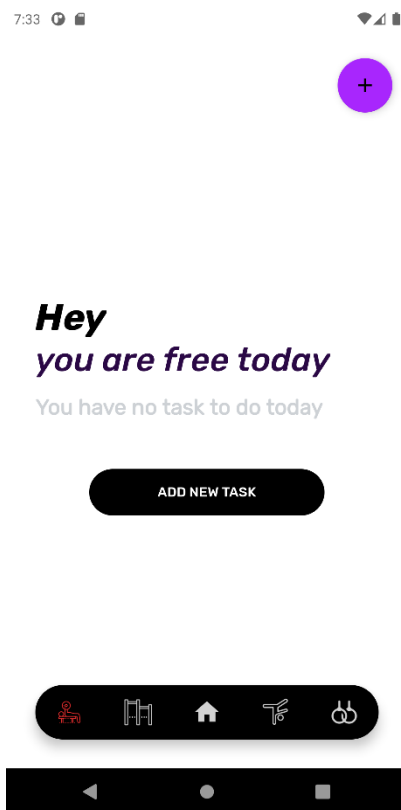
Atgādinājumu var atrast lietotāja sadaļā, sākuma ekrāna sānu navigācijas joslā. Atgādinājumi ir domāti, lai atgādinātu lietotājam piecelties no darba vietas vai vienkārši izkustēties. Aktivitātē ir divas pogas pirmā no tām ir ‘OPEN TIME PICKER’, kur lietotājam tiek izvadīts mazs logs ar pulksteni, kurā jāizvēlas laiks, stundās un minūtes, kādā ir jāskan atgādinājumam, pēc tā parādās uz ekrāna laiks kurā parādīsies atgādinājums . Ar otras pogas palīdzību ir iespējams atcelt atgādinājumu, to nospiežot izvada ka atgādinājums ir atcelts un to atceļ. Atgādinājuma aktivitāte izskatās sekojoši(skāt. 6.3.16. att.).



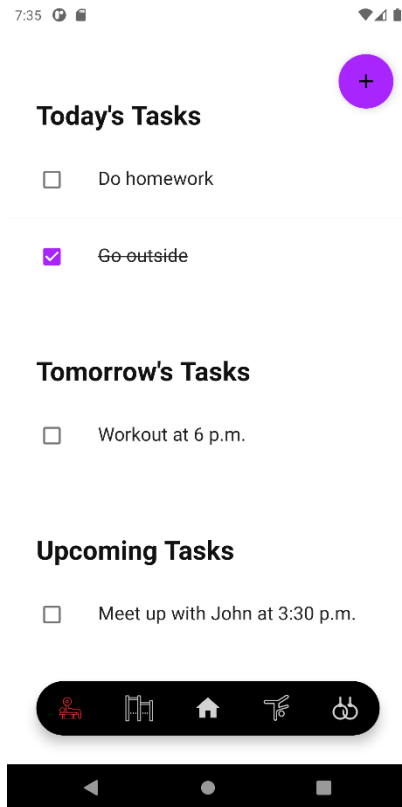
6.3.16. att. Atgādinājuma aktivitāte

Darāmo lietu saraksts

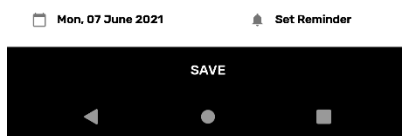
Darāmo lietu sarakstu var atrast lietotāja sadaļā, sākuma ekrāna sānu navigācijas joslā, zem atgādinājuma. Darāmo lietu saraksts ir domāts jebkādiem lietotāja pierakstiem, kas var būt saistīti gan ar treniņiem gan ikdienišķām lietām. Ja darāmo lietu saraksts ir tukšs (skat. 6.3.17. att.), tad tiek izvadīts, ka uzdevumu nav un piedāvā lietotājam pievienot jaunu uzdevumu. Ja lietotājam ir pievienoti uzdevumi tas izskatās šādi (skat. 6.3.18. att.). Uzdevumi tiek dalīti trīs kategorijās – šodienas, rītdienas un gaidāmie- pēc lietotāja ievadīta datuma. Nospiežot uz malējas kastītes, atzīmes ka uzdevums ir izdarīts un pārsvītros tekstu, un ieliks kastīte ķeksi. Uzdevumi, kuriem ir atpakaļ ejošs datums, tiks automātiski dzēsti. Jauna uzdevuma pievienošana (skat. 6.3.19. att.) pēc apraksta ievades, apakša ir poga ar datumu, kurš ir jāievēlas vai pēc noklusējuma būs šodienas datums. Nospiežot uz datuma parādīsies logs ar dienu, mēnesi un gadu, kurā jāizpilda uzdevums. Pēc izvēles var arī pievienot atgādinājumu, kas ir blakus kalendāra pogai. Uzdevumā rediģēšanā (skat. 6.3.20. att.) var mainīt gan aprakstu gan izpildes datumu, kā arī to var manuāli izdzēst.



6.3.17. att. Tukšs darāmo lietu saraksts



6.3.18. att. Aizpildīts darāmo lietu saraksts



6.3.19. Jauna uzdevuma pievienošana

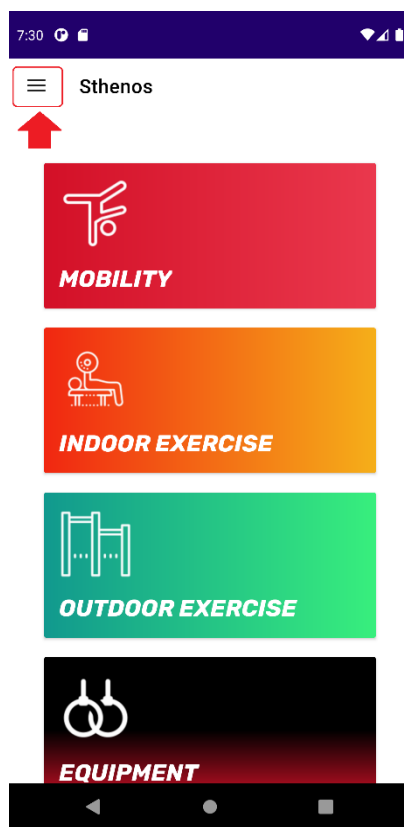


6.3.20. att. Uzdevuma rediģēšana

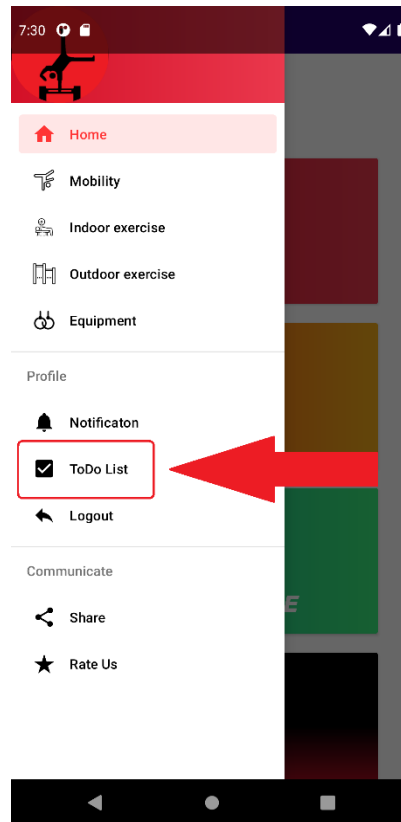
6.4. Testa piemērs

Testa piemēra nolūks ir parādīt darāmo lietu uzdevuma pievienošanu iespējas – apraksta, datuma un atgādinājuma pievienošanu.

Pirmais solis uz vēlamā rezultāta sasniegšanu ir pieteikšanās sistēmā. Kad tas izdarīts, jādodas uz ‘ToDo’ aktivitāti, kas atrodas sākuma lapas augšējā kreisajā stūrī, jāspiež navigācijas joslas poga (skat. 6.4.1. att.). Pēc pogas nospiešanas atveras navigācijas josla, kur jāmeklē ‘ToDo’ aktivitāte zem profila sadaļas (skat. 6.4.2. att.).

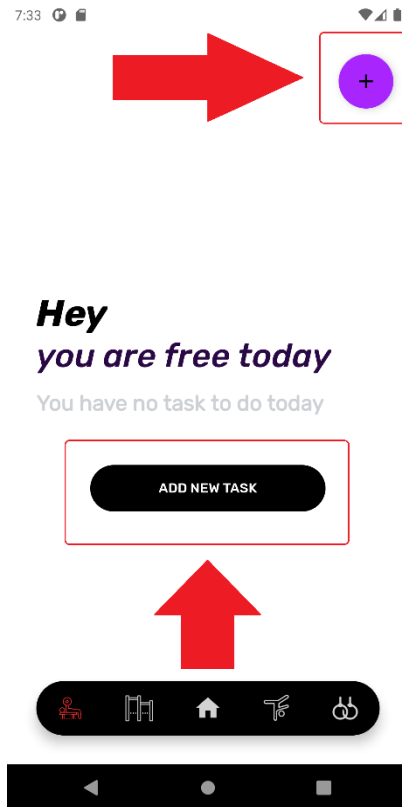


6.4.1. att. Navigācijas poga iezīmēta sarkanā kvadrātā



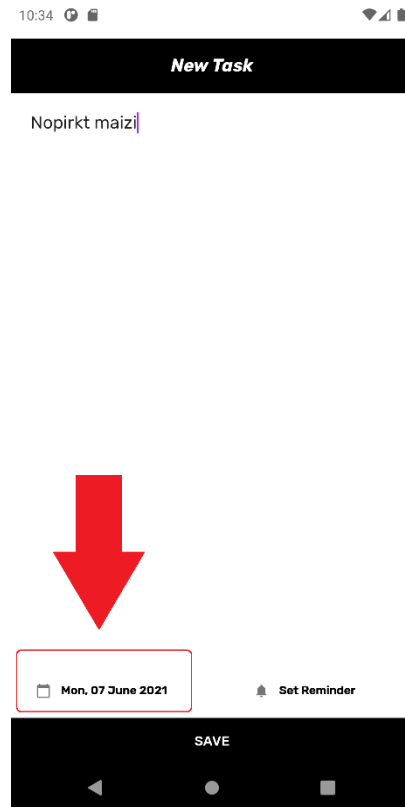
6.4.2. att. Darāmo lietu saraksta atrašanās vieta navigācijas joslā

Nospiežot ‘ToDo List’ atvēršies jauna aktivitāte darāmo lietu saraksts. Kad ir atvēršies darāmo lietu saraksts, tad jāspiež poga ‘ADD NEW TASK’ vai augšējā labajā stūrī ir apaļa poga ar ‘+’ simbolu (skat. 6.4.3. att.).



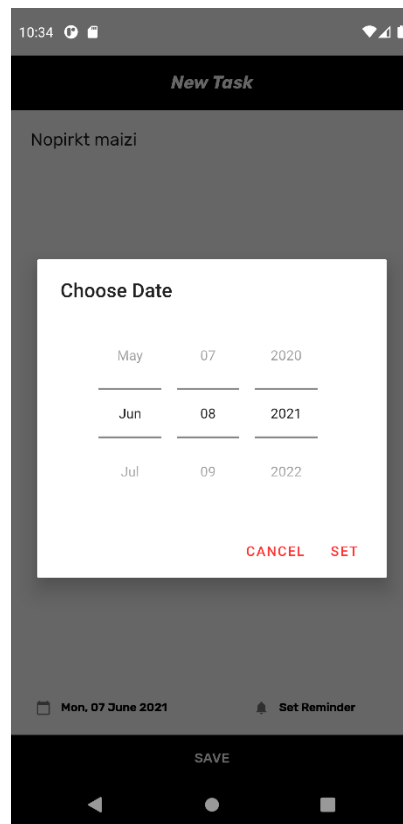
6.4.3. att. Uzdevuma pievienošana

Atverot uzdevuma ievadi, ievadām testa datus, kas ir “Nopirkt maizi ” ar rītdienas datumu. Lai ievadītu datumu jānospiež ikona ar kalendāru un šodienas datumu (skat. 6.4.4. att.).



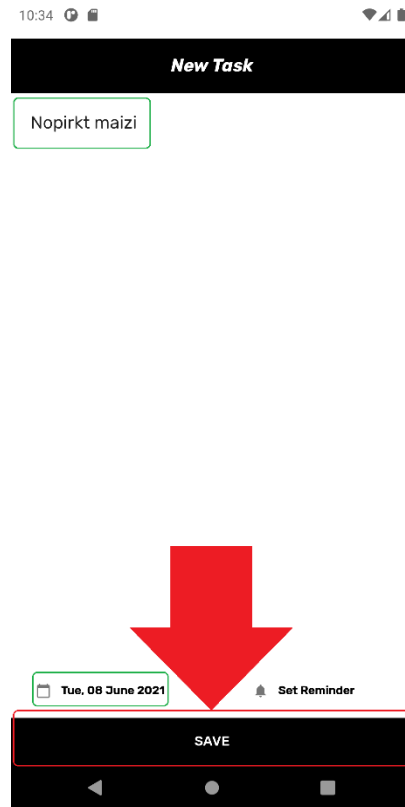
6.4.4. att. datuma izvēles atrašanās vieta

Nospiežot uz kalendāra pogas parādīsies logs ar datuma izvēli, kur jāizvēlas rītdienas datums (skat. 6.4.5. att.).



6.4.5. att. Datuma izvēle

Kad dati ir ievadīti spiež pogu ‘SAVE’, lai saglabātu datus un pievienotu jauno uzdevumu (skat. 6.4.6. att.).



6.4.6. att. Uzdevuma saglabāšana

Ja uzdevums veiksmīgi pievienots, automātiski esam novirzīti atpakaļ uz darāmo lietu saraksta sākumu, kur var redzēt mūsu ievadītos testa datus (skat. 6.4.7. att.).

10:44



Tomorrow's Tasks



Nopirkt maizi



6.4.7. att. Veiksmīgi pievienots jauns uzdevums

NOBEIGUMS

Kvalifikācijas darba mērķis ir sasniegts. Tika izveidota mobilā aplikācija ar kuras palīdzību lietotājiem, kas vairākas stundas pavada sēžot pie datora, un nekustās, tagad ir iespējams ātri un ērti atrast vingrinājumus, kuri palīdzēs palikt aktīvam. Kā arī aplikāciju var izmantot ne tikai, lai izstaipītos, bet gan arī trenētos un attīstītu savu fizisko ķermeni.

Darba mērķis apzinoties šobrīdējo situāciju pasaule ar pandēmiju, šāda veida aplikācija ir nepieciešama tā dēļ, ka liela daļa pasaules sēž mājās un neizkustās pietiekami labi, kas var pasliktināt gan fizisko stāvokli, gan emocionālo stāvokli. Fiziskā aktivitāte ir ļoti nepieciešama, lai cilvēks varētu būt vesels un justies laimīgs. Lietotājs var viegli un ērti uzstādīt atgādinājumus, lai neaizmirstu izkustēties no darba vietas vai vienkārši no sava dīvāna. Tiek piedāvāta opcija arī veidot darāmo lietu sarakstu, ko var izmantot, lai pierakstītu treniņu laikus vai citas ikdienišķas lietas, kas palīdzēs lietotājam būt produktīvam, taču neobligāti tiem ir jābūt uzdevumiem tas var arī būt lietas, ko lietotājs vēlas sasniegt noteiktā laika periodā saistībā ar fizisko izskatu, personīgajiem rekordiņiem vai citas sadzīves lietas.

Izstrādājot kvalifikācijas darbu ieguvu daudz jaunu zināšanu saistība ar mobilo aplikāciju izstrādi, ko noteikti vēl os izmantot arī nākotnē. Nostiprināju arī eksistējošas zināšanas programmēšanas valodas, datubāzes un dizaina veidošanā. Guvu praksi projekta plānošana, skicēšana un veidošanā. Visu pieredzi, ko guvu kvalifikācijas darba izstrādes laika palīdzēs man turpmākajā dzīvē.

INFORMĀCIJAS AVOTI

1. Spraudņa repozitorija - <https://github.com/Vincent-Loi/android-drawable-importer-intellij-plugin>
2. Spraudņa uzstādīšana - <https://stackoverflow.com/questions/60412216/android-drawable-importer-plugin-not-working-in-android-studio-3-6/60795443>
3. Funkcijas, notikumi, metodes, uzskati utt. - <https://developer.android.com/docs>
4. Pogas ar apaļiem galiem veidošana pamācībā - <https://medium.com/@addeeandra/androidstarter-1-i-want-a-round-button-636bc5553d6d>
5. Gradienta veidošanas pamācībā - <https://www.youtube.com/watch?v=v8fDkKB-Vz0>
6. Pieslēgšanās ekrāna dizains pamācībā - https://www.youtube.com/watch?v=C_TEugAIMHA
7. Reģistrācijas ekrāna dizains pamācībā - <https://www.youtube.com/watch?v=aU8dWySoMOU>
8. Sākuma ekrāna dizaina pamācībā - <https://www.youtube.com/watch?v=8oXGXefm4go>
9. Navigāciju joslu pamācības –
 - 9.1. <https://www.youtube.com/watch?v=HwYENW0RyY4>
 - 9.2. <https://www.youtube.com/watch?v=m1RV0HPuBW0>
 - 9.3. <https://www.youtube.com/watch?v=lt6xbth-yQo>
10. ‘Splash screen’ veidošanas pamācība - <https://www.youtube.com/watch?v=JLIFqqnSNmg>
11. Pārstrādātāj Skata režģa izkārtojuma pamācībā - https://www.youtube.com/watch?v=cYjX6_TL_EA
12. Hronometra veidošanas dizains - <https://www.youtube.com/watch?v=gqn7HqTnOPA&t=607s>
13. Peldošas navigācijas joslas dizaina pamācībā - <https://www.youtube.com/watch?v=ePEjNoH-Dlw>
14. Darāmo lietu saraksts – <https://blog.frsarker.com/java/create-a-todo-task-app-using-java-in-android.html>
15. PHP - <https://www.youtube.com/watch?v=X8oD4q3XtQQ>
16. Atgādinājums –

- 16.1. <https://developer.android.com/jetpack/androidx>
- 16.2. <https://developer.android.com/reference/android/app/Notification.Builder>
- 16.3. <https://developer.android.com/reference/android/app/Notification>
- 16.4. <https://www.youtube.com/watch?v=yrpimdBRk5Q>